

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE  
UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À  
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE  
À L'OBTENTION DE LA  
MAÎTRISE EN GÉNIE ÉLECTRIQUE  
M.ING.

PAR  
MARC LAUZON

RÉALISATION D'UN ÉGALISEUR À RETOUR D'ÉTAT AVEC  
DÉCODEUR INTÉGRÉ

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE, 23 FÉVRIER 2001

© droits réservés de Marc Lauzon

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

- M. Jean Belzile, directeur de mémoire  
Département de génie électrique à l'École de technologie supérieure
- M. David Haccoun, professeur  
Département de génie électrique à l'École polytechnique de Montréal
- M. Claude Thibeault, professeur  
Département de génie électrique à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE PRÉSENTATION DEVANT JURY ET UN PUBLIC

LE 31 JANVIER 2001

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

# RÉALISATION D'UN ÉGALISEUR À RETOUR D'ÉTAT AVEC DÉCODEUR INTÉGRÉ

Marc Lauzon

## Sommaire

Les communications à travers un canal de type micro-ondes exigent, entre autres, l'utilisation du codage et de l'égalisation dans le but d'éliminer les effets destructeurs du bruit blanc et de l'interférence intersymbole (ou interférence multi-chemins). Les récepteurs numériques actuels sont ainsi généralement constitués d'un égaliseur et d'un décodeur qui fonctionnent d'une façon indépendante l'un de l'autre.

Ce mémoire démontre qu'il est possible de réaliser dans un circuit programmable, un récepteur qui inclut une structure particulière d'égaliseur à retour d'état à l'intérieur duquel un décodeur à seuil est intégré. Ce récepteur est réalisé entièrement sous forme de portes logiques avant d'être traduit dans un format binaire de programmation par un compilateur.

Des tests sur le récepteur placé dans un environnement constitué d'un générateur de bruit blanc gaussien et d'un simulateur de propagation multi-chemins permettent de conclure que la réalisation de l'égaliseur à retour d'état avec décodeur intégré est fonctionnelle. Certaines modifications doivent cependant être encore apportées au récepteur pour lui permettre d'améliorer davantage ses performances.

## **REMERCIEMENTS**

Je tiens à remercier monsieur Jean Belzile Ph.D. pour son aide et son support apportés tout au long des deux dernières années. Sans lui, la réalisation de ce mémoire ne serait pas possible. J'aimerais également remercier le personnel du Laboratoire de communication et d'intégration de la micro-électronique pour leur support au niveau logiciel et informatique. Enfin, j'aimerais remercier monsieur Sani Amr El Kholy B.ing. et Souha Al-Sawda M.ing. pour leur collaboration dans l'intégration du sujet de ce mémoire dans la radio programmable.

## TABLE DES MATIÈRES

	Page
SOMMAIRE .....	ii
REMERCIEMENTS .....	iii
LISTE DES TABLEAUX .....	vii
LISTE DES FIGURES .....	viii
LISTE DES ABRÉVIATIONS ET DES SIGLES .....	x
INTRODUCTION .....	1
CHAPITRE 1 : SYSTÈMES DE COMMUNICATION .....	3
1.1 Description générale d'un système de communication .....	3
1.2 Codage de canal .....	5
1.2.1 Le codage convolutionnel .....	5
1.2.2 Le décodage à seuil .....	8
1.3 La modulation en quadrature de phase (QPSK) .....	17
1.4 Modélisation du canal de transmission .....	21
1.5 L'égalisation .....	22
1.5.1 L'égaliseur linéaire .....	22
1.5.2 L'égaliseur à retour d'état .....	23
1.5.3 L'égaliseur linéaire complexe et l'égaliseur à retour d'état complexe .....	25
1.6 Les algorithmes d'adaptation .....	25
1.6.1 L'algorithme d'adaptation LMS .....	26
1.6.2 L'algorithme d'adaptation CMA .....	27
1.6.3 L'algorithme d'adaptation MCMA .....	28
1.7 Conclusion .....	29

CHAPITRE 2 : L'ÉGALISEUR À RETOUR D'ÉTAT AVEC DÉCODEUR	
INTÉGRÉ .....	30
2.1 Introduction .....	30
2.2 Architecture de l'égaliseur à retour d'état avec décodeur intégré .....	30
2.2.1 Le filtre avant et le filtre arrière .....	32
2.2.2 Le filtre à réponse impulsionnelle infinie complexe .....	33
2.2.3 Le modulateur .....	34
2.2.4 Le quantificateur .....	35
2.2.5 Le décodeur à seuil .....	37
2.2.5.1 Performances du décodeur à seuil en quantification douce .....	41
2.3 Exemple de performances de l'égaliseur à retour d'état avec décodeur intégré .....	42
2.4 Conclusion .....	43
CHAPITRE 3 : RÉALISATION DE L'ÉGALISEUR À RETOUR D'ÉTAT	
AVEC DÉCODEUR INTÉGRÉ .....	44
3.1 Introduction .....	44
3.1.1 Le récepteur .....	45
3.1.1.1 Le circuit programmable XLA4085 .....	45
3.1.1.2 Les horloges .....	47
3.2 Le filtre avant .....	49
3.3 Le filtre arrière .....	52
3.4 Le filtre à réponse impulsionnelle infinie complexe .....	53
3.5 Le quantificateur .....	54
3.6 Le décodeur à seuil à quantification douce modifié .....	57
3.7 Le modulateur .....	61
3.8 La ligne à délai complexe de trente-cinq (35) coups d'horloge .....	62
3.9 La somme complexe .....	63
3.10 L'adaptation .....	63
3.11 Conclusion .....	69

CHAPITRE 4 : MESURES DES PERFORMANCES .....	71
4.1 Introduction .....	71
4.2 Mesures de performances .....	73
4.3 Conclusion .....	84
CONCLUSION .....	86
BIBLIOGRAPHIE .....	90
ANNEXE A : LES SCHÉMAS LOGIQUES DE RÉALISATION .....	93

## LISTE DES TABLEAUX

	Page
Tableau 1-1 Coefficients de la représentation quadratique d'une modulation QPSK	19
Tableau 2-1 Représentation quadratique de l'information et de la parité .....	34
Tableau 2-2 Signification des poids de fiabilité selon l'axe .....	37
Tableau 3-1 Fréquence des horloges de l'égaliseur .....	47
Tableau 3-2 Intervalle de valeurs des coefficients du filtre avant en simulation .....	49
Tableau 3-3 Intervalle de valeurs des coefficients du filtre arrière en simulation .....	52
Tableau 3-4 Quantification de l'information selon l'intervalle sur l'axe réel .....	55
Tableau 3-5 Table de Karnaugh de l'opérateur addmin à deux opérandes A et B .....	59
Tableau 3-6 La quantification linéaire d'un symbole incluant le conjugué .....	65
Tableau 3-7 La quantification non linéaire d'un symbole .....	66
Tableau 3-8 Pas d'égalisation en fonction du nombre de bits .....	69



## LISTE DES FIGURES

	Page
Figure 1-1 Structure générale d'un système de communication .....	4
Figure 1-2 Codeur convolutionnel pour un code (2,1,3) .....	6
Figure 1-3 Schéma bloc d'un ensemble codeur/décodeur à seuil, code (2,1,6) .....	16
Figure 1-4 Phaseur pour une modulation QPSK .....	18
Figure 1-5 Schéma de principe d'une modulation QPSK .....	20
Figure 1-6 Schéma de principe d'une démodulation QPSK .....	20
Figure 1-7 Modèle de canal .....	21
Figure 1-8 Densité spectrale du bruit blanc en fonction de la fréquence .....	22
Figure 1-9 Schéma bloc d'un filtre transversal .....	23
Figure 1-10 Égaliseur à retour d'état .....	24
Figure 2-1 Schéma bloc de l'égaliseur avec décodeur intégré .....	31
Figure 2-2 Schéma bloc du filtre avant .....	32
Figure 2-3 Schéma bloc du filtre arrière .....	33
Figure 2-4 Schéma bloc du filtre RII complexe .....	33
Figure 2-5 Poids et régions du quantificateur pour l'information et la parité .....	36
Figure 2-6 Probabilité d'erreur en fonction du rapport $E_b/N_0$ .....	41
Figure 2-7 Courbes W sans bruit .....	42
Figure 3-1 Schéma bloc simplifié du récepteur .....	44
Figure 3-2 Schéma bloc de la réalisation de l'égaliseur avec décodeur intégré .....	48
Figure 3-3 Schéma bloc du quantificateur de l'information .....	56
Figure 3-4 Schéma bloc de la réalisation des équations $B^{1B}$ à $B^{8B}$ .....	59
Figure 3-5 Schéma bloc de la réalisation de l'adaptation .....	63
Figure 3-6 Schéma bloc du calcul d'un coefficient complexe .....	67

Figure 4-1 Schéma bloc de l'environnement de test .....	71
Figure 4-2 Taux d'erreur en fonction de $E_b/N_0$ sans évanouissement .....	74
Figure 4-3 Constellation avant l'égaliseur (creux 34dB à 70MHz, sans bruit) .....	75
Figure 4-4 Constellation après l'égaliseur (creux 34dB à 70MHz, sans bruit) .....	76
Figure 4-5 Courbes W sans bruit .....	78
Figure 4-6 Taux d'erreur en fonction de $E_b/N_0$ (creux 28dB à 70MHz) .....	80
Figure 4-7 Constellation avant l'égaliseur (creux 34dB, $E_b/N_0$ 16.6dB) .....	81
Figure 4-8 Constellation après l'égaliseur (creux 34dB, $E_b/N_0$ 16.6dB) .....	82
Figure 4-9 Constellation avant l'égaliseur (creux 34dB, $E_b/N_0$ 14.7dB) .....	83
Figure 4-10 Constellation après l'égaliseur (creux 34dB, $E_b/N_0$ 14.7dB) .....	84

## LISTE DES ABRÉVIATIONS ET DES SIGLES

DFE	Decision Feedback Equalizer
FPGA	Field Programmable Gate Arrays
LACIME	Laboratoire de communications et d'intégration de la micro-électronique
ETS	École de technologie supérieure
T	Durée d'un symbole, s
$s_0(t)$	Signal électrique associé au bit codé 0
$s_1(t)$	Signal électrique associé au bit codé 1
M	Nombre de symboles
v	Nombre de bits à la sortie du codeur
b	Nombre de bits à l'entrée du codeur
K	Longueur de contrainte
m	Mémoire du code
u	La séquence d'information
v	La sortie du codeur
g	La fonction génératrice
G	La matrice génératrice
R	Le taux de codage
e	La séquence d'erreurs
r	La séquence reçue
H	La matrice de contrôle de parité
J	Le nombre de sommes orthogonales de contrôle
s	La séquence des bits de syndrome
$\hat{u}$	La séquence d'information estimée
$\hat{e}$	La séquence d'erreurs estimées

AM	Modulation d'amplitude
FM	Modulation de fréquence
PM	Modulation de phase
ASK	Amplitude-Shift Keying
FSK	Frequency-Shift Keying
PSK	Phase-Shift Keying
QPSK	Modulation en quadrature de phase
QAM	Modulation en quadrature d'amplitude
$s_n(t)$	Signaux représentant la modulation en quadrature de phase
$\theta_n$	Angle de phase
$p_n$	Coefficient du cosinus
$q_n$	Coefficient du sinus
$t$	Temps, s
$c(\tau : t)$	Fonction de transfert de la propagation multi-chemins
$\delta$	Fonction direct
$n(t)$	Valeur du bruit en fonction du temps
$f$	Fréquence
$N_0$	Densité spectrale du bruit
$r(t)$	Signal à la sortie du canal
$s(t)$	Signal à l'entrée du canal
$x_k$	Symbole à l'entrée du filtre avant à l'instant $k$
$y_k$	Symbole à la sortie du filtre avant à l'instant $k$
$c_n$	Coefficient de l'égaliseur
$z_k$	Somme des symboles du filtre avant et du filtre arrière à l'instant $k$
$v_k$	Symbole à la sortie du filtre arrière à l'instant $k$
$\hat{u}_k$	Décision du décodeur à l'instant $k$
MSE	Erreur quadratique moyenne
LMS	Least Mean Square
CMA	Constant Modulus Algorithm

MCAM	Modified Constant Modulus Algorithm
$J_k$	Fonction de coût à l'instant $k$
$E$	Espérance mathématique
$e_k$	Erreur à l'instant $k$
$\Delta$	Pas d'égalisation
$\gamma$	Rayon de Godard
$a_k$	Symbole transmit à l'instant $k$
RII	Réponse impulsionnelle infinie
$I_k$	Information à l'instant $k$
$P_k$	Parité à l'instant $k$
$\hat{I}_k$	Information estimée à l'instant $k$
$s_k$	Symbole à la sortie du filtre à réponse impulsionnelle infinie à l'instant $k$
$A$	Le nombre de poids de la fiabilité
$d$	La distance entre les poids de la fiabilité
$E_N$	Énergie du symbole
$a_i$	Le poids de la fiabilité
$r_i$	La limite d'une région de quantification
$\mathbf{p}$	La matrice des positions à additionner dans le codeur
$B$	Équation de sommes de contrôle
A/N	Analogique à numérique
DQT	Digital Quadrature Tuner
CLB	Configurable Logic Blocks
rx_baud_clk	Horloge au taux de symbole
rx_2xbaud_clk	Horloge au double du taux de symbole
syst_clk	Horloge du système
M_clk	Horloge maître
$\Lambda$	Opérateur addmin à deux entrées
EDI	Égaliseur à retour d'état avec décodeur intégré

## INTRODUCTION

L'utilisation sans cesse grandissante des systèmes de communication numérique est possible et facilitée grâce au développement de la micro-électronique. Cependant, selon le canal de transmission, ces systèmes sont sujets au bruit thermique et aux interférences intersymboles causées, entre autres, par la propagation multi-chemins du signal.

Plusieurs techniques efficaces ont été proposées pour faire face à ces problèmes. Parmi tant d'autres, on retrouve le codage de canal pour protéger le signal contre le bruit thermique et l'égalisation pour éliminer les interférences intersymboles. Les récepteurs numériques actuels sont ainsi généralement conçus avec un égaliseur et un décodeur qui fonctionnent d'une façon indépendante l'un de l'autre.

Récemment, en 1999, une nouvelle architecture d'égaliseur fut proposée par Ladouceur [10]. Cette recherche proposait d'insérer partiellement un décodeur à seuil dans la boucle de rétroaction d'un égaliseur à retour d'état (en anglais «Decision Feedback Equalizer» ou DFE). Les performances de cette nouvelle approche furent évaluées en simulation et les résultats étaient concluants.

L'objectif de ce mémoire est de réaliser, dans un circuit programmable FPGA («Field Programmable Gate Arrays»), un égaliseur à retour d'état avec un décodeur intégré pour ensuite déterminer ces performances dans un environnement de test. L'égaliseur doit être implémenté dans une radio programmable réalisée précédemment au Laboratoire de communication et d'intégration de la micro-électronique (LACIME) de l'École de technologie supérieure (ETS). Le taux de symbole visé est 4 Msymboles par seconde.

Le mandat de l'auteur était donc, dans un premier temps, de comprendre l'architecture proposée par Ladouceur et, ensuite, de l'adapter pour l'intégrer dans le récepteur de la radio programmable en tenant compte des limites du circuit programmable. Ces limites sont principalement : le nombre de cellules programmables disponibles à l'intérieur du circuit et le temps de propagation maximal entre deux cellules programmables en fonction de la fréquence des horloges requises.

Ce mémoire est divisé en quatre (4) chapitres. Le premier chapitre présente une introduction sur les systèmes de communication numérique. Le second chapitre explique en détail l'architecture de l'égaliseur à retour d'état avec décodeur intégré réalisée dans ce mémoire. Le troisième chapitre présente la réalisation de l'architecture sous forme de portes logiques. La quantification utilisée pour chacun des composants et le nombre de cellules logiques programmables nécessaires sont également donnés. Enfin, le dernier chapitre présente les performances du récepteur dans différentes conditions de bruit et d'évanouissement (i.e. dans un milieu de propagation multi-chemins).

## CHAPITRE 1

### SYSTÈMES DE COMMUNICATION

#### 1.1 Description générale d'un système de communication

Un système de communication typique est présenté à la figure 1-1 [1]. Ce système est composé de trois blocs principaux : l'émetteur, le canal et le récepteur. L'émetteur est constitué du codeur de source, du codeur de canal, du modulateur et du filtre de mise en forme. Le récepteur regroupe le filtre de blanchiment, l'égaliseur, le démodulateur, le décodeur de canal et le décodeur de source.

L'information peut être continue ou discrète. Cette information est transformée en une séquence de mots binaires par le codeur de source. Par exemple, en téléphonie, une information continue comme la voix est transformée en valeurs discrètes par un convertisseur analogique à numérique.

La séquence de mots binaires que l'on appelle *séquence d'information* est présentée au codeur de canal. Le rôle du codeur de canal est d'ajouter de la redondance contrôlée à la séquence d'information. La séquence d'information devient alors une *séquence codée*. La redondance d'information est utilisée au récepteur dans le but de contrer les effets du bruit et des interférences produits lors de la transmission du signal à travers le canal.



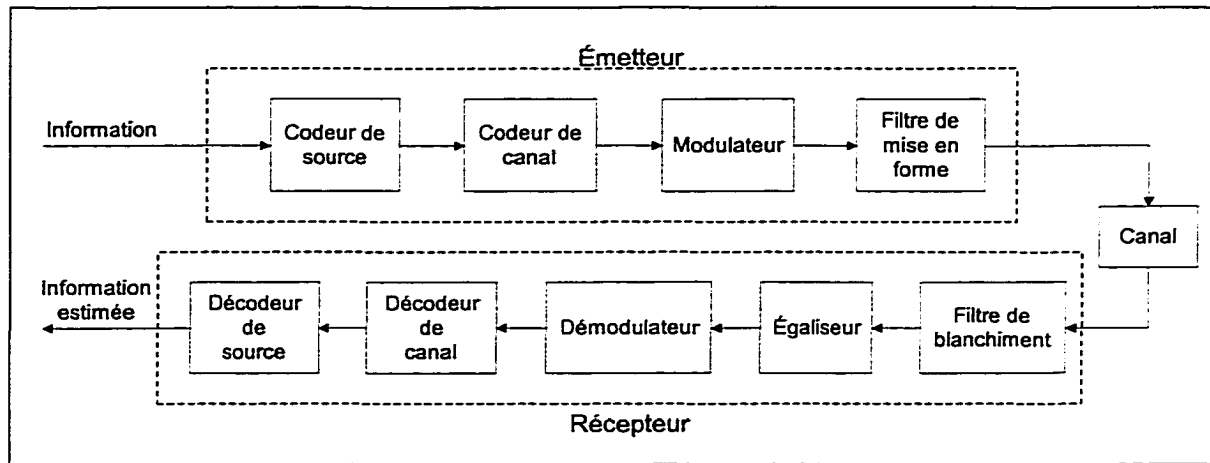


Figure 1-1 Structure générale d'un système de communication.

Avant de transmettre les séquences codées sur un canal physique tel une ligne de téléphone, un lien radio en haute fréquence, un lien de télémétrie, un lien micro-onde ou satellite, la séquence discrète doit être transformée en un signal électrique continu. La fonction du modulateur est donc d'associer un ensemble de bits de la séquence codée à une forme d'onde de durée  $T$  secondes que l'on appelle *symbole*. Le symbole peut représenter un ou plusieurs bits de la séquence codée. Par exemple, pour une séquence codée ne contenant que les bits codés 0 et 1, le modulateur peut associer le signal  $s_0(t)$  au bit codé 0 et le signal  $s_1(t)$  au bit codé 1. Dans ce cas, le symbole ne représente qu'un bit de la séquence codée. En général, si le symbole représente  $b$  bits de la séquence codée, le modulateur doit pouvoir transmettre  $M = 2^b$  signaux distincts. Lorsque  $M > 2$ , la littérature parle de modulation  $M$ -aire.

Le filtre de mise en forme ajoute au signal modulé une résistance contre les interférences en concentrant l'énergie du signal au centre de chaque symbole. Il limite également la largeur de bande du signal à transmettre. Typiquement, le filtre de mise en forme est un filtre à cosinus surélevé [1][3].

Chaque canal physique possède des caractéristiques de bruit et d'interférence qui lui sont propres. Par contre, une chose sûre est que tous dégraderont le signal d'une façon ou d'une autre. Au récepteur, le signal corrompu est d'abord filtré et égalisé pour éliminer les effets des interférences intersymboles. Ensuite, le démodulateur analyse les formes d'onde reçues et produit la *séquence reçue*. Le décodeur de canal estime la séquence codée à l'émetteur grâce au code et à la redondance présente dans la séquence reçue. On parle alors de *séquence estimée*. Enfin, le décodeur de source reconvertit la séquence estimée en un format utilisable pour le destinataire (convertisseur numérique à analogique pour un haut-parleur par exemple).

## 1.2 Le codage de canal

Le codage de canal est utilisé depuis longtemps comme une méthode de correction d'erreurs spécialement pour les transmissions limitées en puissance. La combinaison du codage et de la modulation donne le gain de codage. Deux types de code sont généralement utilisés pour obtenir ce gain : les codes blocs et les codes convolutionnels. Le codage convolutionnel fut introduit par Elias [4] en 1955 comme une alternative moins complexe à réaliser et aussi efficace que les codes blocs. C'est pourquoi ce type de codage fera l'objet de discussion dans les prochaines sections.

### 1.2.1 Le codage convolutionnel

La notation et la présentation de cette section et de la section suivante suivent celles utilisées par Lin et Costello [2]. Le codeur convolutionnel est un système linéaire composé d'une mémoire finie (un registre à décalage) et d'additionneurs modulo 2 (portes ou-exclusif). Le codeur est décrit par son nombre  $v$  de bits à la sortie, par son nombre  $b$  de bits à l'entrée, par sa longueur de contrainte  $K$  et par ses connexions. Le *taux de codage*  $R$  est donné par le rapport  $b/v$ . La *longueur de contrainte*  $K$  est celle du registre à décalage. La *mémoire du code*  $m$  est le nombre de bits précédents nécessaires

pour coder le bit présent. L'emplacement des connexions détermine les changements d'état possibles parmi les  $2^m$  états. Les fonctions génératrices du codeur sont données par sa réponse à l'impulsion. Par exemple, supposons le codeur convolutionnel avec un code  $(v=2, b=1, m=3)$  comme présenté à la figure 1-2.

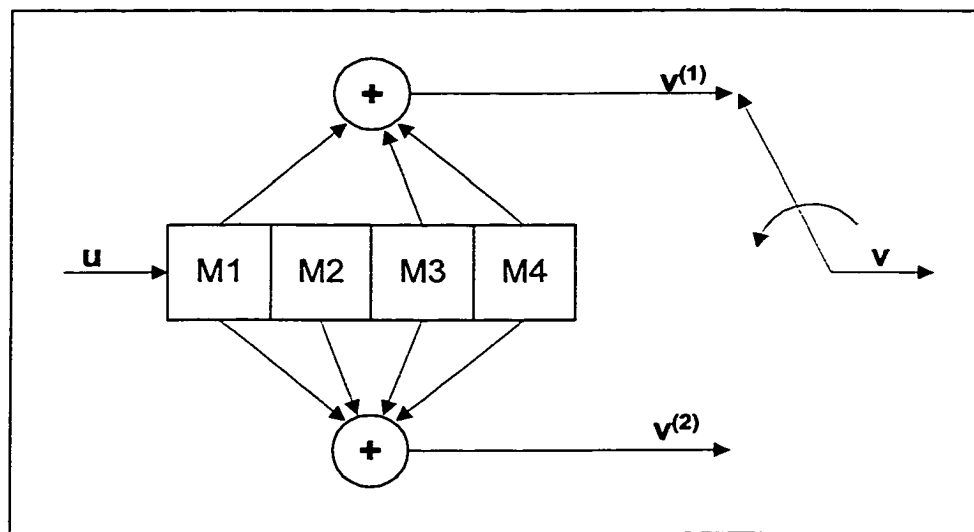


Figure 1-2 Codeur convolutionnel pour un code (2,1,3).

La séquence d'information  $\mathbf{u} = (u_0, u_1, u_2, \dots)$  entre dans le codeur un bit à la fois. Les séquences résultantes à la sortie de chaque additionneur sont  $\mathbf{v}^{(1)} = (v_0^{(1)}, v_1^{(1)}, v_2^{(1)}, \dots)$  et  $\mathbf{v}^{(2)} = (v_0^{(2)}, v_1^{(2)}, v_2^{(2)}, \dots)$ . La sortie du codeur  $\mathbf{v} = (v_0^{(1)}, v_0^{(2)}, v_1^{(1)}, v_1^{(2)}, v_2^{(1)}, v_2^{(2)}, \dots)$  est le résultat du multiplexage des sorties de chaque additionneur. La séquence  $\mathbf{u} = (1, 0, 0, \dots)$  donne la réponse à l'impulsion du codeur. Par le fait même, les deux fonctions génératrices  $\mathbf{g}^{(1)}$  et  $\mathbf{g}^{(2)}$  du codeur de la figure 1-2 sont :

$$\mathbf{g}^{(1)} = (1 \ 0 \ 1 \ 1),$$

$$\mathbf{g}^{(2)} = (1 \ 1 \ 1 \ 1).$$

Les fonctions génératrices  $g^{(1)}$  et  $g^{(2)}$  sont souvent présentées en format octal (i.e.  $g^{(1)}=13$  et  $g^{(2)}=17$ ). Le codeur étant un système linéaire, il est possible de trouver directement la séquence de sortie en faisant la convolution de l'entrée  $u$  avec chacune des fonctions génératrices  $g^{(1)}$  et  $g^{(2)}$  de la façon suivante :

$$v^{(1)} = u * g^{(1)}, \quad (1-1a)$$

$$v^{(2)} = u * g^{(2)}. \quad (1-1b)$$

L'opérateur  $*$  représente la convolution discrète. Il faut noter que toutes les opérations sont de type modulo 2. Ainsi, l'opération de convolution pour  $l \geq 0$  donne :

$$v_l^{(j)} = \sum_{i=0}^m u_{l-i} g_i^{(j)} = u_l g_0^{(j)} \oplus u_{l-1} g_1^{(j)} \oplus \dots \oplus u_{l-m} g_m^{(j)} \quad j = 1, 2, \dots \quad (1-2)$$

où  $u_{l-i}$  vaut 0 pour tous les  $l < i$ . Pour notre exemple de la figure 1-2, les sorties  $v^{(1)}$  et  $v^{(2)}$  deviennent alors :

$$v_l^{(1)} = u_l \oplus u_{l-2} \oplus u_{l-3},$$

$$v_l^{(2)} = u_l \oplus u_{l-1} \oplus u_{l-2} \oplus u_{l-3}.$$

On peut également trouver la séquence de sortie grâce à la matrice génératrice  $G$ . Cette dernière est constituée des fonctions génératrices entrelacées. Avec les fonctions  $g^{(1)}$  et  $g^{(2)}$  par exemple, la matrice  $G$  est générée comme suit :

$$G = \begin{bmatrix} g_0^{(1)} g_0^{(2)} & g_1^{(1)} g_1^{(2)} & g_2^{(1)} g_2^{(2)} & \dots & g_m^{(1)} g_m^{(2)} \\ g_0^{(1)} g_0^{(2)} & g_1^{(1)} g_1^{(2)} & \dots & g_{m-1}^{(1)} g_{m-1}^{(2)} & g_m^{(1)} g_m^{(2)} \\ & \cdot & & \cdot & \cdot \\ & & \cdot & & \cdot \\ & & & \cdot & \cdot \\ & & & & \cdot \end{bmatrix}. \quad (1-3)$$

Les espaces vides de la matrice  $G$  représentent des zéros. Chaque rangée de  $G$  est identique à la rangée précédente mais décalée de  $v$  positions vers la droite. La matrice  $G$  est également semi-infinie puisque la séquence d'information  $u$  est d'une longueur arbitraire. Si  $u$  a une longueur de  $L$  bits, alors  $G$  a  $L$  rangés et  $\frac{v}{b}(m+1)$  colonnes. Enfin, la séquence  $v$  est de longueur  $\frac{v}{b}(m+L)$  et elle est trouvée avec le calcul matriciel suivant :

$$v = uG. \quad (1-4)$$

### 1.2.2 Le décodage à seuil

Le décodage à seuil ou logique majoritaire fut appliqué aux codes convolutionnels par Massey [5] en 1963. Le décodeur à seuil diffère des autres décodeurs, comme le décodeur de Viterbi et le décodeur séquentiel, dans sa façon de prendre la décision finale [2]. Le décodeur à seuil prend sa décision finale sur une longueur de contrainte de la séquence reçue contrairement aux autres décodeurs qui basent leur décision sur toute la séquence reçue. Cette particularité rend le décodeur à seuil plus simple à réaliser. Par contre, ses performances sont moindres que celles du décodeur de Viterbi et du décodeur séquentiel [2].

Pour mieux expliquer le décodage par logique majoritaire, supposons un code  $R = \frac{1}{2}$  avec les fonctions génératrices  $g^{(1)} = (1 \ 0 \ 0 \ \dots)$  et  $g^{(2)} = (g_0^{(2)}, g_1^{(2)}, g_2^{(2)}, \dots)$ . Tel que montré par l'équation 1-3, la matrice génératrice est :

$$G = \begin{bmatrix} 1 & g_0^{(2)} & 0 & g_1^{(2)} & 0 & g_2^{(2)} & \dots & 0 & g_m^{(2)} \\ & & 1 & g_0^{(2)} & 0 & g_1^{(2)} & \dots & 0 & g_{m-1}^{(2)} & 0 & g_m^{(2)} \\ & & & \cdot & & \cdot & & \cdot & \cdot & & \cdot \\ & & & & \cdot & & \cdot & & \cdot & & \cdot \end{bmatrix}. \quad (1-5)$$

A partir de 1-1a et 1-1b, les équations de codage sont :

$$\mathbf{v}^{(1)} = \mathbf{u} * \mathbf{g}^{(1)} = \mathbf{u}, \quad (1-6)$$

$$\mathbf{v}^{(2)} = \mathbf{u} * \mathbf{g}^{(2)}. \quad (1-7)$$

La séquence d'information est représentée par le vecteur  $\mathbf{u}$ . Un code semblable est nommé code systématique car l'information non codée est incluse dans le mot de code à transmettre.

Si le mot de code  $\mathbf{v} = \mathbf{uG}$  est transmis à travers un canal binaire symétrique, la séquence binaire reçue  $\mathbf{r}$  est décrite par :

$$\mathbf{r} = (r_0^{(1)}, r_0^{(2)}, r_1^{(1)}, r_1^{(2)}, r_2^{(1)}, r_2^{(2)}, \dots) = \mathbf{v} + \mathbf{e} \quad (1-8)$$

où  $\mathbf{e} = (e_0^{(1)}, e_0^{(2)}, e_1^{(1)}, e_1^{(2)}, e_2^{(1)}, e_2^{(2)}, \dots)$  est la séquence d'erreurs du canal. Un bit d'erreur  $e_i^{(j)}$  vaut «1» si et seulement si le bit reçu  $r_i^{(j)}$  n'est pas le même que le bit transmis  $v_i^{(j)}$ . Autrement dit, lorsqu'il y a une erreur. On peut diviser la séquence reçue  $\mathbf{r}$  en deux parties. La première partie  $\mathbf{r}^{(1)}$  représente la séquence d'information reçue et la seconde partie  $\mathbf{r}^{(2)}$  représente la parité reçue :

$$\mathbf{r}^{(1)} = (r_0^{(1)}, r_1^{(1)}, r_2^{(1)}, \dots) = \mathbf{v}^{(1)} + \mathbf{e}^{(1)} = \mathbf{u} + \mathbf{e}^{(1)}, \quad (1-9)$$

$$\mathbf{r}^{(2)} = (r_0^{(2)}, r_1^{(2)}, r_2^{(2)}, \dots) = \mathbf{v}^{(2)} + \mathbf{e}^{(2)} = \mathbf{u} * \mathbf{g}^{(2)} + \mathbf{e}^{(2)}. \quad (1-10)$$

Conséquemment,  $\mathbf{e}^{(1)}$  est la séquence d'erreurs sur l'information et  $\mathbf{e}^{(2)}$  est la séquence d'erreurs sur la parité. Le décodeur doit déterminer si la séquence d'informations reçue contient des erreurs de transmission sans connaître  $\mathbf{u}$  et  $\mathbf{e}$ . Lorsqu'une erreur est détectée, le décodeur doit localiser l'erreur et la corriger. Pour ce faire, il calcule le syndrome de la séquence reçue  $\mathbf{s}$  à l'aide de la matrice de contrôles de parité  $\mathbf{H}$  :



On dit qu'un bit d'erreur  $e_j$  est vérifié par une somme de contrôles si ce bit est inclus dans la somme. Un ensemble de  $J$  sommes de contrôles est orthogonal sur le bit d'erreur  $e_j$  si :

- Chaque somme de contrôles vérifie  $e_j$ .
- Aucun autre bit d'erreur n'est vérifié par plus d'une somme de contrôles.

Ainsi, pour un ensemble de  $J$  sommes orthogonales de contrôles, le bit d'erreur  $e_j$  est estimé grâce à la règle de majorité du décodeur. Cette règle stipule que l'on estime  $e_j$  à «1» si et seulement si plus de la moitié des sommes orthogonales de contrôles sur  $e_j$  ont une valeur de «1». Si la moitié ou moins des sommes orthogonales de contrôles ont une valeur de «1» alors le bit d'erreur  $e_j$  vaut «0».

Pour mieux illustrer le calcul des sommes de contrôles sur  $e_0^{(1)}$ , prenons l'exemple d'un code systématique  $R=1/2$  avec  $\mathbf{g}^{(2)} = (1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1)$  [2]. Selon l'équation 1-13,  $e_0^{(1)}$  ne peut affecter les bits de syndrome que sur une longueur de contrainte ( $K=7$ ). Définissons  $[s]_6$  comme étant égal à  $(s_0, s_1, s_2, \dots, s_6)$ . Avec la même notation et en calculant la matrice de contrôles de parité, l'équation 1-13 devient :



$$[\mathbf{s}]_6 = [\mathbf{e}]_6 [\mathbf{H}^T]_6$$

$$[\mathbf{s}]_6 = [\mathbf{e}]_6 \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ & 1 & 1 & 0 & 0 & 1 & 0 \\ & 1 & 0 & 0 & 0 & 0 & 0 \\ & & 1 & 1 & 0 & 0 & 1 \\ & & 1 & 0 & 0 & 0 & 0 \\ & & & 1 & 1 & 0 & 0 \\ & & & 1 & 0 & 0 & 0 \\ & & & & 1 & 1 & 0 \\ & & & & & 1 & 0 \\ & & & & & & 1 \\ & & & & & & & 1 \end{bmatrix}$$

(1-14)

En transposant les deux côtés de l'équation 1-14, le résultat est :

$$[\mathbf{s}]_6^T = \begin{bmatrix} 11 \\ 1011 \\ 001011 \\ 00001011 \\ 1000001011 \\ 001000001011 \\ 10001000001011 \end{bmatrix} [\mathbf{e}]_6^T$$

(1-15)

Les colonnes paires de  $\mathbf{H}^T$  correspondantes à la séquence d'erreurs sur la parité forment une matrice identité. Sachant cela, l'équation 1-15 peut s'écrire :

$$[\mathbf{s}]_6^T = \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \end{bmatrix} = \begin{bmatrix} 1 \\ 11 \\ 011 \\ 0011 \\ 10011 \\ 010011 \\ 1010011 \end{bmatrix} \begin{bmatrix} e_0^{(1)} \\ e_1^{(1)} \\ e_2^{(1)} \\ e_3^{(1)} \\ e_4^{(1)} \\ e_5^{(1)} \\ e_6^{(1)} \end{bmatrix} + \begin{bmatrix} e_0^{(2)} \\ e_1^{(2)} \\ e_2^{(2)} \\ e_3^{(2)} \\ e_4^{(2)} \\ e_5^{(2)} \\ e_6^{(2)} \end{bmatrix} . \quad (1-16)$$

La matrice de l'équation 1-16 qui multiplie la séquence d'erreurs sur l'information est appelée triangle de parité du code. Il est bon de noter que la première colonne du triangle de parité est la fonction génératrice  $g^{(2)}$ . Les autres colonnes représentent aussi  $g^{(2)}$  mais décalées d'une position et tronquées vers le bas à chaque changement de colonne.

Le triangle de parité peut être utilisé pour déterminer un ensemble de sommes orthogonales de contrôles. Pour ce faire, il faut tenir compte de deux choses importantes :

- Aucun bit de syndrome ne peut être utilisé plus d'une fois dans une somme orthogonale de contrôles.
- Le bit de parité est le seul qui doit être vérifié plus d'une fois.

L'exemple ci-dessus montre qu'il y a quatre bits de syndrome qui vérifie  $e_0^{(1)}$ . Il est, par conséquent, impossible d'obtenir plus de quatre sommes orthogonales de contrôles sur  $e_0^{(1)}$ . Les sommes orthogonales de contrôles sont illustrées par le triangle de parité de l'équation 1-17 :

$$\begin{array}{rcl}
& \rightarrow & 1 \\
& \rightarrow & 1 \boxed{1} \\
& & 0 \ 1 \ 1 \\
& & 0 \ 0 \ 1 \ 1 \\
& \rightarrow & 1 \ 0 \ 0 \boxed{1} \boxed{1} \\
& & 0 \ 1 \ 0 \ 0 \ 1 \ 1 \\
& \rightarrow & 1 \ 0 \boxed{1} \ 0 \ 0 \boxed{1} \boxed{1}
\end{array} . \tag{1-17}$$

Les flèches indiquent les bits de syndrome qui vérifient  $e_0^{(1)}$ . Ces bits de syndrome sont choisis comme étant les sommes de contrôles sur  $e_0^{(1)}$ . Les «1» encadrés indiquent quels bits d'erreur sur l'information autre que  $e_0^{(1)}$  sont aussi vérifiés. Les équations de sommes orthogonales de contrôles se résument ainsi :

$$\begin{array}{ll}
s_0 = e_0^{(1)} & + e_0^{(2)}, \\
s_1 = e_0^{(1)} + e_1^{(1)} & + e_0^{(2)}, \\
s_4 = e_0^{(1)} & + e_3^{(1)} + e_4^{(1)} + e_0^{(2)}, \\
s_6 = e_0^{(1)} & + e_2^{(1)} + e_5^{(1)} + e_6^{(1)} + e_0^{(2)}.
\end{array} \tag{1-18}$$

Ce code est auto-orthogonal puisque chaque somme de contrôles est représentée par un seul bit de syndrome et non par une somme de bits de syndrome. La même démarche peut être appliquée pour calculer les sommes de contrôles sur  $e_1^{(1)}$ . L'estimation de  $e_0^{(1)}$  est d'abord utilisée pour modifier les bits de syndrome ( $s_1, s_2, s_3, \dots, s_6$ ). Un nouveau bit de syndrome  $s_7$  est calculé. Un nouvel ensemble de sommes orthogonales de contrôles est formé et la même règle de décodage est appliquée.

Le diagramme bloc complet d'un ensemble codeur/décodeur à seuil pour le même code est présenté à la figure 1-3. Les opérations du décodeur peuvent se résumer comme suit :

1. La première longueur de contrainte de bits de syndrome ( $s_0, s_1, s_2, \dots, s_6$ ) est calculée.
2. Un ensemble de quatre (4) sommes orthogonales de contrôles sur  $e_0^{(1)}$  est formé à partir des bits de syndrome calculés précédemment.
3. Les quatre (4) sommes de contrôles sont présentées à une logique de majorité qui produit un «1» à sa sortie si trois (3) ou quatre (4) des sommes sont à «1». Si la sortie de la logique de majorité est «1» alors le bit d'erreur  $e_0^{(1)}$  estimé vaut «1» et le bit d'information reçu  $r_0^{(1)}$  doit être corrigé (i.e. une erreur est détectée). Le bit d'information reçu  $r_0^{(1)}$  est corrigé par une addition modulo 2 avec le bit d'erreur estimé  $\hat{e}_0^{(1)}$ . La sortie de la logique de majorité est également renvoyée en rétroaction et soustraite de chaque bit de syndrome affecté.
4. Le bit d'information estimé  $\hat{u}_0 = r_0^{(1)} + \hat{e}_0^{(1)}$  sort du décodeur. Le registre de syndrome est décalé d'une position vers la droite et le prochain bloc de bits reçus (i.e.  $r_7^{(1)}$  et  $r_7^{(2)}$ ) sont présentés au décodeur. Le prochain bit de syndrome  $s_7$  est calculé et envoyé à l'étage le plus à gauche du registre de syndrome.
5. Le registre de syndromes contient maintenant les bits de syndrome modifiés par la sortie de la logique majoritaire et le bit de syndrome  $s_7$ . Le décodeur répète les étapes 2,3 et 4 et estime  $e_1^{(1)}$ . Tous les bits d'erreur et les bits d'information suivants seront estimés de la même façon.

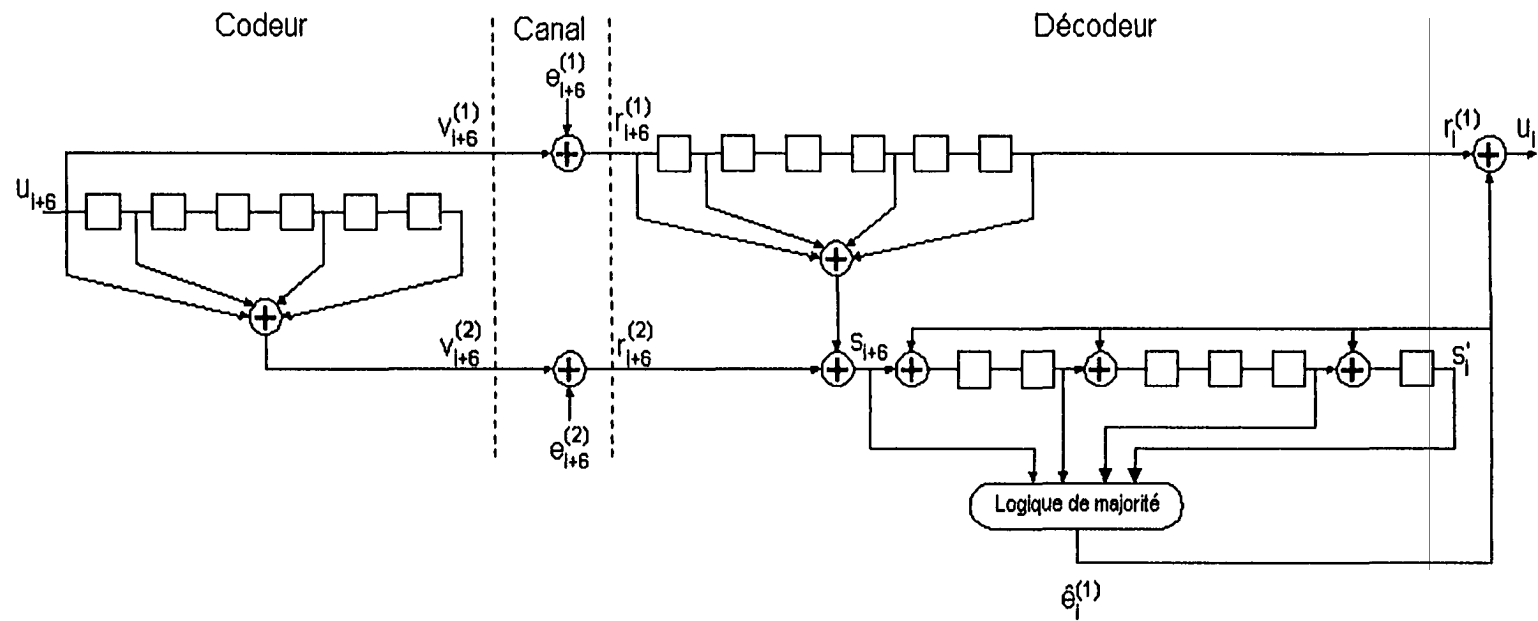


Figure 1-3 Schéma bloc d'un ensemble codeur/décodeur à seuil pour un code  $(2,1,6)[2]$ .

### 1.3 La modulation en quadrature de phase (QPSK)

La modulation est un processus qui modifie une caractéristique quelconque d'une porteuse en fonction d'un message. Les modulations analogiques de base sont : la modulation d'amplitude (AM), la modulation de fréquence (FM) et la modulation de phase (PM). Les équivalents numériques de ces modulations sont : « amplitude-shift keying (ASK) », « frequency-shift keying (FSK) » et « phase-shift keying (PSK) ». Les modulations numériques se caractérisent par le fait qu'elles transmettent des symboles composés de  $M$  valeurs discrètes et séparés d'un intervalle de temps régulier  $T$ . Le choix du type de modulation numérique pour une application dépend, en général, du taux d'erreur désiré, de la largeur de bande disponible et de la complexité de réalisation [3].

La modulation binaire, par exemple, qui possède seulement deux symboles, est plus simple à réaliser, donne de bonnes performances d'erreur mais demande une largeur de bande importante. D'un autre côté, la modulation  $M$ -aire transmet un message d'une longueur de  $b$  bits avec chaque symbole et est, par conséquent, appropriée pour des hauts taux de transmission. L'utilisation de la largeur de bande est alors plus efficace. Nous n'étudierons ici que la modulation  $M$ -aire PSK et plus particulièrement la modulation en quadrature de phase (QPSK) car c'est celle qui fut choisie dans le cadre du projet à cause de sa complexité réduite et de ses performances acceptables.

Le principe de modulation en quadrature que nous expliquerons ici pour la modulation  $M$ -aire PSK, peut être généralisé pour inclure une modulation d'amplitude au même titre qu'une modulation de phase. Comme nous le verrons plus loin, en PSK, la phase et les composantes quadratiques du signal ne sont pas indépendantes. Les valeurs des composantes quadratiques sont choisies de façon à maintenir l'enveloppe du signal constante, ce qui est une caractéristique fondamentale du PSK. Si cette contrainte est enlevée de façon à ce que les composantes quadratiques soient indépendantes une de l'autre, le résultat est une modulation en quadrature d'amplitude (QAM). De plus, pour

le cas où le nombre de signaux possibles est quatre ( $M=4$ ), comme en QPSK, alors ce dernier est identique au 4-QAM.

La modulation M-aire PSK peut être représentée par l'ensemble de signaux suivants :

$$s_n(t) = A \cos(\omega_c t + \theta_n) \quad 0 \leq t \leq T \quad (1-19)$$

où les  $M$  symboles sont exprimés par un ensemble d'angles de phase séparés uniformément :

$$\theta_n = \frac{(2n-1)\pi}{M} \quad n = 1, 2, \dots, M. \quad (1-20)$$

La différence entre deux angles de phase adjacents est de  $2\pi / M$ . Pour une modulation en quadrature de phase, cette différence est de  $\pi/2$ . Une deuxième façon plus pratique de représenter les signaux M-aire PSK est le phaseur qui affiche l'amplitude du signal et son angle de phase. La figure 1-4 montre un exemple de phaseur pour une modulation QPSK. Les points sur le phaseur forment la constellation du signal. Comme l'indique la figure 1-4, chaque signal a une amplitude  $A$ , ce qui a pour effet de créer un cercle de rayon  $A$ . Les régions de décision sont montrées par les lignes pointillées. Le seuil de décision du démodulateur est situé au centre de deux phases successives de façon à ce que le démodulateur prenne la bonne décision si la phase reçue se trouve dans l'intervalle  $\pm\pi/M$  de la phase émise.

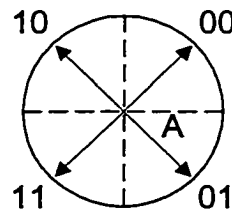


Figure 1-4 Phaseur pour une modulation QPSK

On peut aussi représenter les signaux M-aire PSK avec la représentation en quadrature. Celle-ci utilise l'expansion trigonométrique pour décomposer le signal en sinus et en cosinus :

$$s_n(t) = A[p_n \cos(\omega_c t) + q_n \sin(\omega_c t)] \quad 0 \leq t \leq T \quad (1-21)$$

où :

$$p_n = \cos(\theta_n), \quad (1-22)$$

$$q_n = \sin(\theta_n). \quad (1-23)$$

L'ensemble des coefficients  $p_n$  et  $q_n$  est donné dans le tableau 1-1 pour une modulation QPSK. Il faut noter aussi que dans ce cas-ci et pour le phaseur, le code de Gray a été utilisé pour assigner les bits aux signaux. Ainsi, le code binaire ne diffère que d'un bit entre les signaux adjacents.

Tableau 1-1

Coefficients pour la représentation quadratique d'une modulation QPSK

Valeur des bits	$P_n$	$q_n$	Angle de phase
00	1	1	$\pi/4$
10	-1	1	$3\pi/4$
11	-1	-1	$5\pi/4$
01	1	-1	$7\pi/4$

Cette représentation est pratique également pour expliquer comment générer du QPSK à partir de données binaires (la séquence codée par exemple). La figure 1-5 présente le schéma bloc d'une modulation en quadrature de phase.



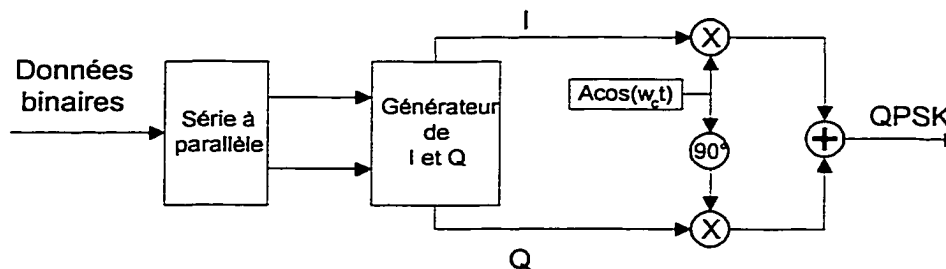


Figure 1-5 Schéma de principe d'une modulation QPSK [3].

Les données binaires sont d'abord converties en parallèle. Le générateur prend les données en parallèle et les associe à une paire de coefficients  $p_n$  et  $q_n$  qui sont les coefficients des signaux I et Q. Les signaux I et Q sont ensuite multipliés par un cosinus et un sinus respectivement. Enfin, les résultats de chaque multiplication sont additionnés pour former le signal QPSK.

La démodulation se fait généralement par détection cohérente. Cette dernière requiert la génération d'une référence locale au récepteur. Le recouvrement de la porteuse est également nécessaire pour retrouver la phase. La figure 1-6 montre un exemple de démodulateur en quadrature de phase.

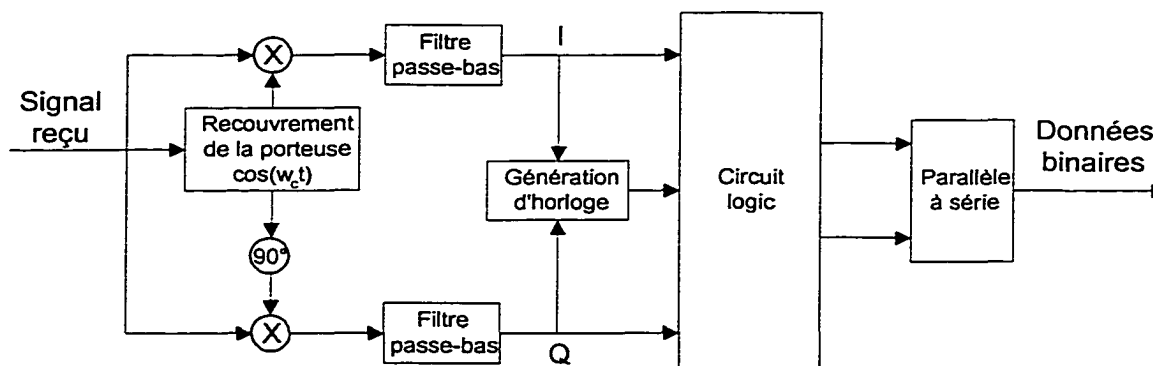


Figure 1-6 Schéma de principe d'une démodulation QPSK [3].

### 1.4 Modélisation du canal de transmission

Le canal est le lien entre l'émetteur et le récepteur. Ce lien est un milieu physique sujet au bruit thermique et aux interférences intersymboles (ISI). En radiotélécommunication, les interférences intersymboles sont le résultat d'une propagation multi-chemins. Le modèle mathématique le plus couramment utilisé pour représenter un canal sujet au bruit thermique et aux interférences intersymboles est présenté à la figure 1-7 [1].

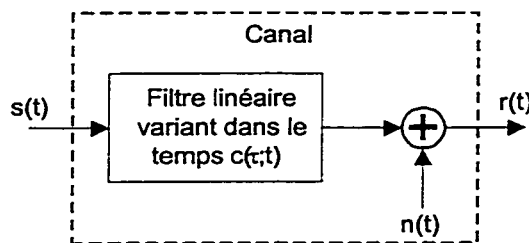


Figure 1-7 Modèle de canal.

La propagation multi-chemins est modélisée par un filtre linéaire variant dans le temps ayant la fonction de transfert suivante :

$$c(\tau; t) = \sum_{k=1}^L a_k(t) \delta(\tau - \tau_k). \quad (1-24)$$

Le paramètre  $a_k(t)$  représente un facteur d'atténuation variant dans le temps pour les  $L$  chemins de propagation. En résumé, on peut décrire la propagation multi-chemins comme une transmission à travers plusieurs canaux ayant des gains et des délais différents.

Le bruit thermique est représenté par l'addition d'une valeur  $n(t)$  à la sortie du filtre. Pour les systèmes de communication, le bruit thermique est modélisé par le bruit blanc

gaussien. Le bruit blanc a ainsi une densité spectrale  $N_0$  uniforme pour toutes les fréquences comme montré à la figure 1-8.

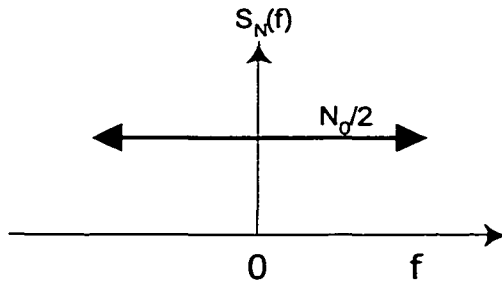


Figure 1-8 Densité spectrale du bruit blanc en fonction de la fréquence.

La sortie du canal  $r(t)$  est donc la convolution de l'entrée  $s(t)$  avec la fonction de transfert  $c(\tau;t)$  du filtre linéaire variant dans le temps auquel on additionne le bruit. L'équation 1-25 illustre le modèle mathématique du canal :

$$r(t) = \sum_{k=1}^L a_k(t)s(t - \tau_k) + n(t). \quad (1-25)$$

## 1.5 L'égalisation

L'égalisation est utilisée pour combattre les distorsions causées principalement par la propagation multi-chemins [7]. Le concept de l'égalisation est basé sur un filtre qui compense les distorsions sur la phase et sur l'amplitude. En d'autres mots, l'égalisation peut être vue comme le filtre inverse du filtre linéaire variant dans le temps du canal.

### 1.5.1 L'égaliseur linéaire

L'égaliseur linéaire constitué d'un filtre transversal est le choix le plus commun lorsqu'on parle d'égalisation d'un signal à cause de sa simplicité et de sa versatilité [3].

Le filtre transversal est composé d'une ligne à délai où les délais sont séparés généralement de  $T$  secondes (le temps d'un symbole). Chaque étage de la ligne à délai est multiplié par un certain facteur avant d'être additionné ensemble (i.e. addition décimale normale). La sortie de l'égaliseur est le résultat de l'addition. Les facteurs de multiplication que l'on appelle *coefficients* sont calculés de façon à soustraire les effets des interférences des symboles adjacents sur le symbole présent. La figure 1-9 présente un exemple de filtre transversal.

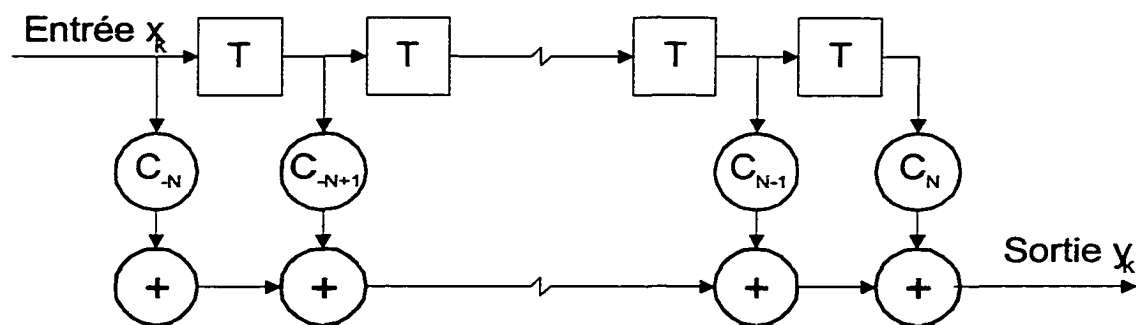


Figure 1-9 Schéma bloc d'un filtre transversal.

En supposant que le filtre a  $(2N+1)$  étages, que les coefficients de chaque étages sont  $c_{-N}, c_{-N+1}, \dots, c_N$  et que l'entrée est  $x_k$  comme présenté à la figure 1-9, il est possible d'exprimer la sortie  $y_k$  sous la forme suivante :

$$y_k = \sum_{n=-N}^N c_n x_{k-n} \quad k = -2N, \dots, 2N. \quad (1-26)$$

### 1.5.2 L'égaliseur à retour d'état

L'égaliseur à retour d'état est la suite logique de l'égaliseur linéaire dans le développement des égaliseurs [3][8]. La figure 1-10 représente la structure de ce type d'égaliseur.

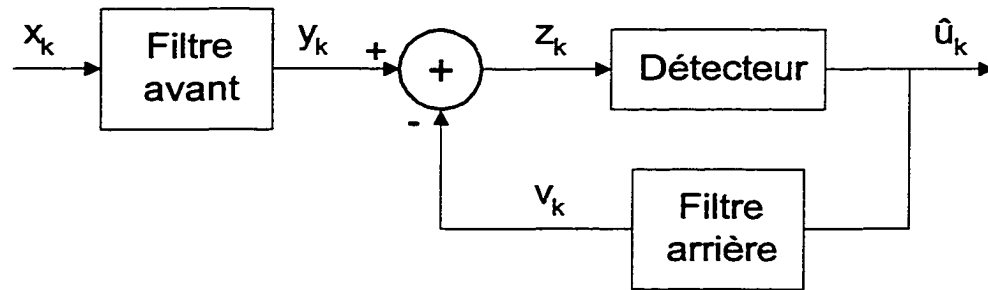


Figure 1-10 Égaliseur à retour d'état.

L'égaliseur à retour d'état est donc composé de trois blocs importants : le filtre avant, le filtre arrière et le détecteur. Le filtre avant est un filtre transversal linéaire qui tente d'éliminer les interférences intersymboles causées par les symboles futurs. Les décisions du détecteur prises à partir du filtre avant sont présentées au filtre arrière. Ce dernier est aussi un filtre transversal linéaire. En supposant que les décisions prises dans le passé soient correctes, le filtre arrière élimine les interférences intersymboles causées par les symboles passés. Le rôle du détecteur est d'éliminer le bruit pour le filtre arrière en déterminant à quelle région appartient  $z_k$ . Le détecteur rend ainsi le DFE plus efficace dans la réduction du bruit.

Maintenant, en supposant que le filtre avant a  $N$  étages et que le filtre arrière a  $M$  étages, la sortie  $z_k$  représentant la somme du filtre avant et du filtre arrière est donné par :

$$z_k = \sum_{n=-N}^0 c_n x_{k-n} + \sum_{n=1}^M c_n \hat{u}_{k-n} \quad k = -2N, \dots, M. \quad (1-27)$$

L'estimation de l'information envoyée  $\hat{u}_k$  est le résultat de la décision du détecteur prise sur  $z_k$ .

### 1.5.3 L'égaliseur linéaire complexe et l'égaliseur à retour d'état complexe

Dans le cas d'un système avec une modulation en quadrature comme le QPSK par exemple, l'égaliseur, qu'il soit linéaire ou à retour d'état, doit être en mesure de traiter les signaux complexes. Les figures 1-9 et 1-10 sont valables également pour des nombres complexes. Les seules différences sont que les lignes à délai sont formées d'une ligne pour la partie réelle et d'une ligne pour la partie imaginaire du signal et que toutes les opérations d'addition et de multiplication sont complexes. Les coefficients de chaque étage sont aussi complexes. La partie réelle des coefficients aide à combattre les interférences intersymboles dans les canaux en phase et à quadrature. La partie imaginaire élimine l'interférence croisée entre les deux canaux.

## 1.6 Les algorithmes d'adaptation

Comme il a été mentionné dans la section 1.4, les caractéristiques du filtre du canal changent avec le temps, l'égaliseur doit être en mesure de s'adapter au canal. On parle alors d'égalisation adaptative.

Pour l'égalisation adaptative, les coefficients de l'égaliseur s'ajustent au fur et à mesure que les caractéristiques du canal changent. L'égaliseur peut se baser sur ces propres décisions pour évaluer les coefficients optimaux selon le canal à un moment précis. Différents algorithmes d'adaptation existent pour calculer les coefficients. Nous verrons brièvement quelques-uns de ces algorithmes dans les sections suivantes. Pour plus d'information, le lecteur est référé aux volumes et articles suivants [1][3][7][9][11][15][17][21][24].

### 1.6.1 L'algorithme d'adaptation LMS

L'algorithme d'adaptation des moindres carrés LMS («Least-mean Square») fut développé à la fin de l'année 1960 par Widrow et Hoff [25]. Cet algorithme tente de trouver les coefficients qui minimiseront l'erreur quadratique moyenne MMSE («Minimum Mean Square Error»).

Par exemple, pour l'égaliseur linéaire de la figure 1-9, nous ajoutons un détecteur à la sortie du sommateur. Ce détecteur prend une décision sur  $y_k$  et donne  $\hat{u}_k$ . La fonction à minimiser  $J$  est l'erreur quadratique moyenne qui se définit comme l'espérance mathématique de l'erreur quadratique :

$$J_k = E(e_k^2) = E(y_k - \hat{u}_k)^2. \quad (1-28)$$

L'erreur quadratique moyenne est alors une fonction quadratique des coefficients puisque  $y_k$  dépend directement de la valeur des  $N$  coefficients. On peut représenter l'erreur quadratique moyenne comme une parabole de  $N$  dimensions et possédant un minimum. Ainsi pour trouver ce minimum, il faut trouver la combinaison de coefficients qui annule la dérivée de l'erreur quadratique moyenne par rapport à chacun des coefficients.

$$\frac{\partial J}{\partial c} = 0 \quad (1-29)$$

Pour solutionner l'équation 1-29, Widrow et Hoff [25] ont démontré qu'il était possible d'utiliser  $\frac{\partial}{\partial c} e_k^2$  à la place de  $\frac{\partial J}{\partial c}$  et ils ont trouvé que :

$$\frac{\partial}{\partial c} e_k^2 = 2e_k \bar{x}_k = 2(y_k - \hat{u}_k) \bar{x}_k. \quad (1-30)$$

Enfin, les coefficients de chacun des  $(2N+1)$  étages sont calculés de façon itérative à l'aide du gradient trouvé en 1-30 et ce à intervalle d'un symbole :

$$\bar{c}_{k+1} = \bar{c}_k + \Delta(y_k - \hat{u}_k)\bar{x}_k . \quad (1-31)$$

Le pas d'égalisation  $\Delta$  est variable et il détermine la vitesse de convergence de l'adaptation. Un grand pas d'égalisation permet une convergence plus rapide mais il y a un risque de non-convergence. Un petit pas d'égalisation assure une convergence vers une solution mais le temps de convergence est plus long.

En terminant, pour un égaliseur complexe le raisonnement est le même et les coefficients complexes se calculent de la façon suivante :

$$\bar{c}_{k+1} = \bar{c}_k + \Delta(y_k - \hat{u}_k)\bar{x}_k^* . \quad (1-32)$$

Chaque variable de l'équation est alors composée d'une partie réelle et d'une partie imaginaire. La différence est que le conjugué de  $x$  est utilisé pour le calcul à chaque itération.

### 1.6.2 L'algorithme d'adaptation CMA

L'algorithme d'adaptation CMA («Constant Modulus Algorithm») fut proposé par Godard [11] en 1980. Cet algorithme tente de trouver les coefficients qui minimiseront un critère appelé la dispersion d'ordre  $p$ . Toujours avec l'exemple de l'égaliseur de la figure 1-9 et pour une dispersion d'ordre 2, la fonction à minimiser  $J$  se résume comme suit :

$$J_k = E\{(\gamma - |y_k|^2)^2\} . \quad (1-33)$$



La constante  $\gamma$  appelée rayon de Godard est calculée comme le rapport de l'espérance mathématique de la norme des symboles transmis  $a_k$  exposant quatre (4) sur l'espérance mathématique de la norme des symboles transmis au carrée :

$$\gamma = \frac{E\{|a_k|^4\}}{E\{|a_k|^2\}}. \quad (1-34)$$

Dans le cas de notre modulation QPSK par exemple, le rayon de Godard est égal à 2. Comme pour l'algorithme LMS, la méthode itérative incluant le gradient de la fonction à minimiser est utilisée pour calculer les coefficients. Après développement, Godard a démontré que les coefficients complexes se calculent de la façon suivante :

$$\bar{c}_{k+1} = \bar{c}_k + \Delta y_k (2 - |y_k|^2) \bar{x}_k^*. \quad (1-35)$$

### 1.6.3 L'algorithme d'adaptation MCMA

L'algorithme d'adaptation MCMA («Modified Constant Modulus Algorithm») fut proposé par Oh et Chin [21] en 1995. La fonction à minimiser est modifiée et ses parties réelle et imaginaire sont séparées. Pour une dispersion d'ordre 2, la fonction à minimiser  $J$  est :

$$J_k = J_{k,R} + J_{k,I}, \quad (1-36)$$

où  $J_{k,R}$  et  $J_{k,I}$  sont respectivement la partie réelle et la partie imaginaire de la fonction coût de la sortie de l'égaliseur  $y_k = y_{k,R} + j \cdot y_{k,I}$  :

$$J_{k,R} = E\{(\gamma_R - |y_{k,R}|^2)^2\}, \quad (1-37)$$

$$J_{k,I} = E\{(\gamma_I - |y_{k,I}|^2)^2\}. \quad (1-38)$$

Les constantes  $\gamma_R$  et  $\gamma_I$  sont calculées ainsi :

$$\gamma_R = \frac{E\{|a_{k,R}|^4\}}{E\{|a_{k,R}|^2\}^2}, \quad (1-39)$$

$$\gamma_I = \frac{E\{|a_{k,I}|^4\}}{E\{|a_{k,I}|^2\}^2}. \quad (1-40)$$

Dans le cas de notre modulation QPSK,  $\gamma_R = \gamma_I = 1$ . Enfin, la méthode itérative incluant le gradient de la fonction à minimiser est toujours utilisée pour calculer les coefficients complexes :

$$\bar{c}_{k+1} = \bar{c}_k + \Delta [y_{k,R}(1 - |y_{k,R}|^2) + j \cdot y_{k,I}(1 - |y_{k,I}|^2)] \bar{x}_k^*. \quad (1-35)$$

## 1.7 Conclusion

Nous avons vu dans ce chapitre les principales composantes de base d'une communication numérique. Nous avons vu que l'information est codée, modulée et filtrée à l'émetteur avant d'être transmise sur le canal. Toutes ces opérations ont pour but de protéger le signal d'information contre le bruit et les interférences que le canal engendre inévitablement. L'information ainsi corrompue est estimée par le récepteur grâce à des opérations de filtrage, d'égalisation, de démodulation et de décodage. Le récepteur optimal est celui qui prend une décision sur l'information reçue qui conduit à une probabilité d'erreur minimale. Le chapitre suivant présente une nouvelle approche dans l'estimation de l'information reçue au récepteur.

## CHAPITRE 2

### L'ÉGALISEUR À RETOUR D'ÉTAT AVEC DÉCODEUR INTÉGRÉ

#### 2.1 Introduction

Ce chapitre présente l'architecture de l'égaliseur à retour d'état avec décodeur intégré qui a été réalisée dans ce mémoire. L'architecture est basée sur celle proposée par Ladouceur en 1999 [10] pour estimer l'information qui a été transmise à partir de l'information reçue au récepteur.

Comme nous l'avons vu au chapitre 1, un bon récepteur doit avoir un égaliseur et un décodeur pour combattre les effets du bruit et des interférences intersymboles. L'architecture proposée combine partiellement un décodeur à seuil à même la boucle d'un égaliseur à retour d'état. Cette combinaison permet d'obtenir une information plus fiable à l'entrée du filtre arrière de l'égaliseur et ainsi de limiter la propagation d'erreurs pour un rapport signal à bruit de faible intensité [10].

#### 2.2 Architecture de l'égaliseur à retour d'état avec décodeur intégré

La figure 2-1 résume l'architecture de l'égaliseur à retour d'état avec décodeur intégré. Les symboles reçus au récepteur sont présentés à l'égaliseur sous forme de nombres complexes. La porteuse a été préalablement enlevée et chaque nombre complexe est formé de la composante en I du symbole sur l'axe réel et de la composante en Q du symbole sur l'axe imaginaire. Les symboles ainsi représentés entrent dans un premier filtre complexe que l'on appelle *filtre avant*. Après avoir été égalisés une

première fois, les symboles sont présentés à un filtre à réponse impulsionnelle infinie complexe (RII) et à une ligne à délai complexe.

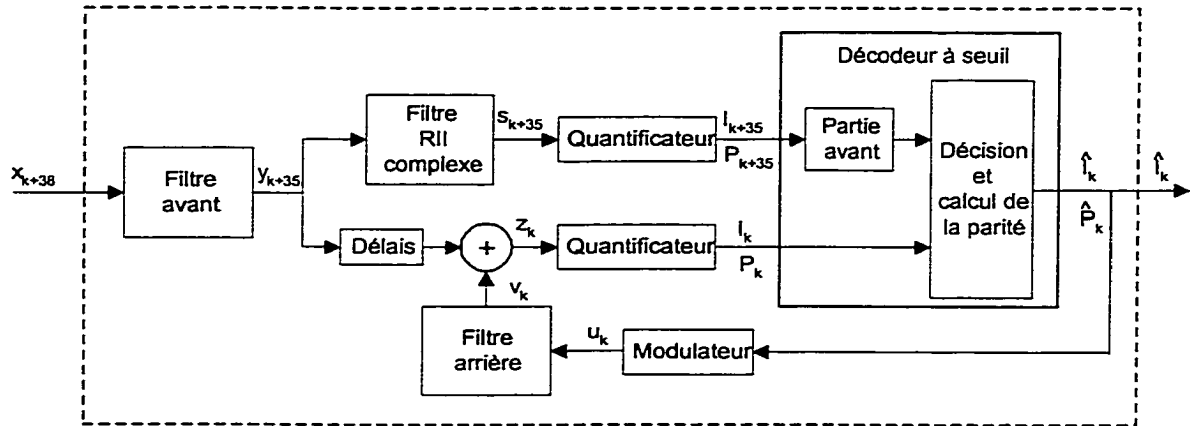


Figure 2-1 Schéma bloc de l'égaliseur avec décodeur intégré.

Examinons, en premier lieu, les symboles égalisés par le filtre avant qui sont présentés à la ligne à délai. La longueur de cette ligne à délai est équivalente à la longueur de contrainte du décodeur à seuil. Une fois les symboles retardés, ils sont additionnés avec la sortie du filtre de rétroaction que l'on appelle *filtre arrière*. Ce filtre utilise à son entrée les décisions prises par le décodeur. Ces décisions ont été remodulées au préalable. Le résultat de la somme est quantifié et présenté au décodeur à seuil.

Dans la seconde branche, celle du filtre RII complexe, les symboles sont filtrés et quantifiés avant d'être présentés à la partie avant du décodeur à seuil. Enfin, le décodeur estime l'information reçue, calcule la parité de l'information estimée et présente l'information estimée à la sortie du récepteur. Les sections suivantes expliquent, en détail, chacun des composants de l'égaliseur à retour d'état avec décodeur intégré.

### 2.2.1 Le filtre avant et le filtre arrière

Le filtre avant est un filtre transversal linéaire complexe comme présenté au chapitre 1. Ce filtre fonctionne à deux fois le taux de symbole. Un tel filtre se nomme filtre par fractionnement du temps. Son principal avantage est qu'il s'adapte mieux aux caractéristiques du canal et qu'il se rapproche ainsi du filtre adapté [9]. Comme le montre la figure 2-2, l'étage central de ce filtre se trouve au dernier étage ce qui permet d'éliminer les interférences des symboles futurs sur le symbole présent.

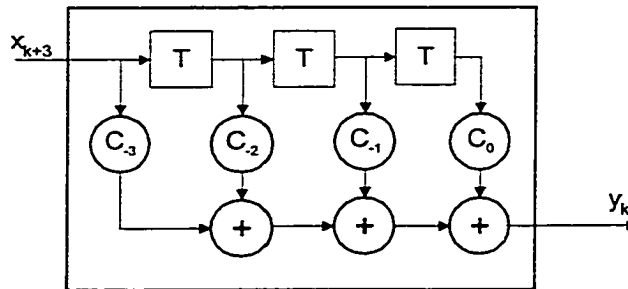


Figure 2-2 Schéma bloc du filtre avant.

Pour le mémoire, la longueur du filtre avant a été fixée à quatre (4) étages pour des raisons d'efficacité acceptable pour une complexité réalisable. La sortie du filtre avant est donc :

$$y_k = \sum_{n=-3}^0 c_n x_{k-n} . \quad (2-1)$$

Le filtre arrière est aussi un filtre transversal linéaire complexe. Il est présenté à la figure 2-3. Ce filtre, par contre, fonctionne au taux de symbole puisqu'il utilise les décisions du décodeur remodulées au taux de symbole. Le filtre arrière élimine les interférences intersymboles causées par les symboles passés.

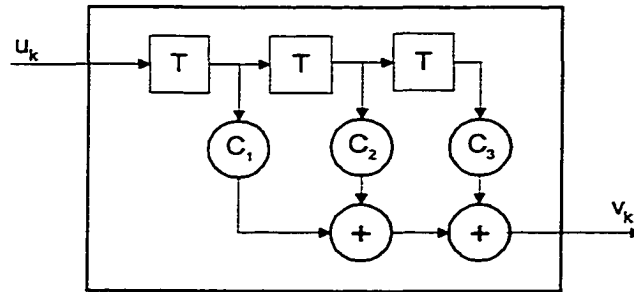


Figure 2-3 Schéma bloc du filtre arrière.

La longueur du filtre arrière est fixée à trois (3) étages parce qu'il a été démontré, pour le canal de Rummier (canal présenté à la section 1.4), que ce nombre d'étages réduit le plus l'erreur quadratique moyenne [8] sur un signal égalisé. La sortie du filtre arrière est calculée comme suit :

$$v_k = \sum_{n=1}^3 c_n u_{k-n} . \quad (2-2)$$

### 2.2.2 Le filtre à réponse impulsionnelle infinie complexe

Ce filtre a été inclus dans l'architecture de l'égaliseur à retour d'état avec décodeur intégré afin de réduire l'interférence post-curseur (ou interférence produite par les symboles passés) sur la partie avant du décodeur à seuil [10]. La figure 2-4 montre le schéma bloc du filtre à réponse impulsionnelle infinie complexe.

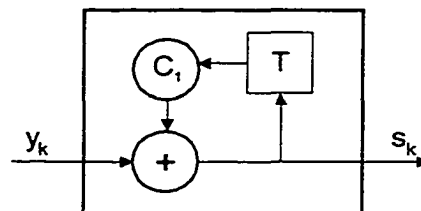


Figure 2-4 Schéma bloc du filtre RII complexe.

Ce filtre peut être vu comme un filtre de rétroaction à un étage seulement et ce pour limiter la propagation d'erreurs. D'ailleurs, le coefficient de ce filtre est le même que celui du premier étage du filtre arrière autrement dit  $c_1$ . Par conséquent, l'équation du filtre est :

$$s_k = \sum_{i=0}^k c_i y_{k-i} . \quad (2-3)$$

### 2.2.3 Le modulateur

Le modulateur ajouté dans la boucle du filtre arrière est, à toute fin utile, le modulateur QPSK présenté dans le chapitre 1 à l'exception qu'il fonctionne en bande de base. Comme nous l'avons vu, ce modulateur a besoin de deux bits codés pour former un symbole. Pour notre système, les deux bits sont le bit d'information et le bit de parité calculé à partir de l'information. Il est à noter que le même principe est utilisé dans le modulateur de l'émetteur. Ce modulateur associe donc l'information estimée à sa composante I correspondante (celle-ci forme la partie réelle du nombre complexe) et la parité estimée à sa composante Q correspondante (celle-là forme la partie imaginaire du nombre complexe). Les symboles ainsi créés sont présentés au filtre arrière. Le tableau 2-1 résume l'association de chaque bit avec son correspondant quadratique.

Tableau 2-1

Représentation quadratique de l'information et de la parité.

Information	Parité	I	Q
0	0	1	1
0	1	1	-1
1	0	-1	1
1	1	-1	-1

### 2.2.4 Le quantificateur

Le rôle du quantificateur est de deux ordres : extraire l'information et la parité du symbole représenté par un signal complexe et leur associer chacun un niveau discret c'est-à-dire une fiabilité. La fiabilité sera utilisée par le décodeur en quantification douce pour lui permettre d'estimer correctement l'information transmise à l'émetteur.

Comme il a été mentionné dans la section précédente, l'information se trouve sur la partie réelle du signal complexe (la composante I) et la parité est sur la partie imaginaire (la composante Q). La partie réelle et la partie imaginaire seront donc traitées séparément.

Le niveau discret sur l'information est choisi parmi un ensemble de niveaux discrets équidistants les uns par rapport aux autres. Ces niveaux représentent le poids de la fiabilité de l'information. La plage de valeurs possibles du signal complexe reçu est divisée en région et le poids de chacune des régions est le point milieu de la région en question. Le poids sur la parité est déterminé de la même façon mais en traitant la partie imaginaire du signal complexe.

Par exemple, en considérant les niveaux transmis  $(\sqrt{E_N}, -\sqrt{E_N})$  au modulateur et pour un nombre A de poids, la distance entre les poids est [15] :

$$d = \frac{\sqrt{E_N} - (-\sqrt{E_N})}{A-1} = \frac{2\sqrt{E_N}}{A-1}. \quad (2-4)$$

Ensuite, les poids  $a_1, a_2, \dots, a_A$  sont calculés de la façon suivante :

$$a_i = -\sqrt{E_N} + d(i-1) \quad i = 1, 2, \dots, A. \quad (2-5)$$



Les limites des régions  $r_1, r_2, \dots, r_{A-1}$  sont déterminées ainsi :

$$r_i = \frac{a_i + a_{i+1}}{2} \quad i = 1, 2, \dots, A-1. \quad (2-6)$$

Dans le cadre de cette thèse, le nombre de poids a été choisi à six(6) et ce pour des raisons de réalisation expliquées au chapitre suivant. La figure 2-5 présente les différents poids et régions utilisés pour l'information et pour la parité en fonction du signal complexe.

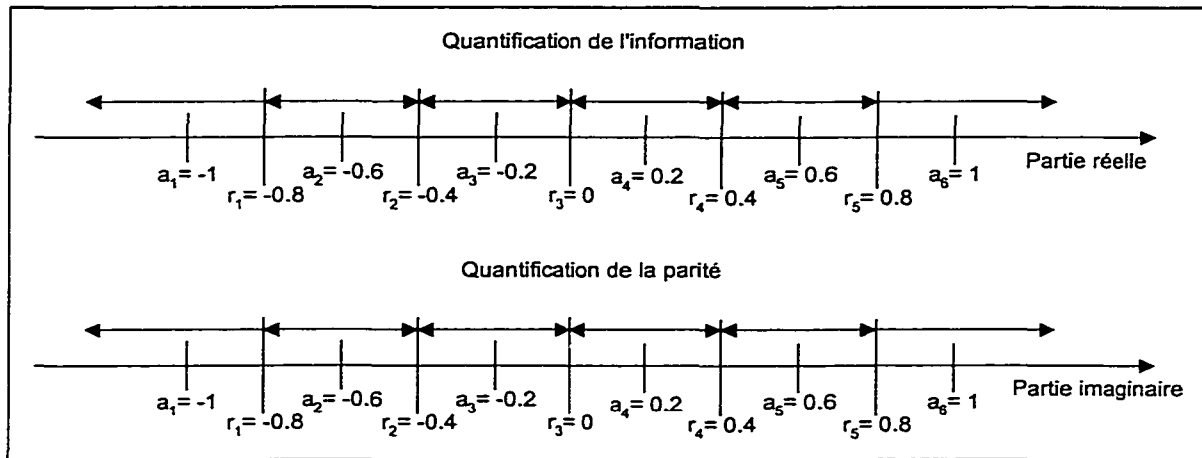


Figure 2-5 Poids et régions du quantificateur pour l'information et la parité.

La signification des poids de la fiabilité en fonction des valeurs sur les axes réel et imaginaire est donnée dans le tableau 2-2.

Tableau 2-2  
Signification des poids de fiabilité selon l'axe.

Intervalle sur la l'axe réel	Signification	Intervalle sur l'axe imaginaire	Signification
$-\infty, -0.8[$	Bit d'information est 1 très fiable	$-\infty, -0.8[$	Bit de parité est 1 très fiable
$[-0.8, -0.4[$	Bit d'information est 1 fiable	$[-0.8, -0.4[$	Bit de parité est 1 fiable
$[-0.4, 0[$	Bit d'information est 1 peu fiable	$[-0.4, 0[$	Bit de parité est 1 peu fiable
$[0, 0.4]$	Bit d'information est 0 peu fiable	$[0, 0.4]$	Bit de parité est 0 peu fiable
$]0.4, 0.8]$	Bit d'information est 0 fiable	$]0.4, 0.8]$	Bit de parité est 0 fiable
$]0.8, \infty$	Bit d'information est 0 très fiable	$]0.8, \infty$	Bit de parité est 0 très fiable

### 2.2.5 Le décodeur à seuil

Le décodeur à seuil utilisé dans l'architecture de la figure 2-1 est le décodeur AAPP («Approximate A Posteriori Probability») en quantification douce, légèrement modifié pour pouvoir être inclus dans la boucle de l'égaliseur à retour d'état. Ce type de décodeur fut présenté en 1991 [13] comme une approximation plus facilement réalisable du décodeur APP («A Posteriori Probability») proposé par Massey [5]. Dans les deux cas, la base du décodage est une logique de majorité comme celle présentée au chapitre 1.

Avant d'expliquer en détail le décodeur à seuil, il est important de présenter le code utilisé dans le cadre de cette thèse, c'est-à-dire le code utilisé au codeur de l'émetteur. Le code choisi est un code systématique auto-orthogonal ( $v=2$ ,  $b=1$ ,  $m=35$ ). La distance minimale de ce code est neuf (9) et sa capacité de correction est quatre (4). Ce code offre un compromis acceptable entre sa complexité de réalisation et ses performances. Les fonctions génératrices du code représentées en octal sont  $g^{(1)} = 400000000000$  et  $g^{(2)} = 402202400061$ . Comme nous avons vu au chapitre 1, l'information fait partie intégrante du mot code et la parité générée avec la fonction  $g^{(2)}$  est incluse aussi dans le mot code. La parité est calculée avec une addition modulo 2 des bits qui se trouvent aux positions 0,7,10,16,18,30,31 et 35 dans le registre à décalage du codeur.

Les équations de sommes de contrôles du décodeur à seuil AAPP sont évidemment basées sur le code utilisé au codeur. Dans le cas d'un code  $R = \frac{1}{2}$  et en définissant une matrice  $p=(p_1, p_2, p_3, \dots, p_J)$  qui contient les positions à additionner pour le calcul de la parité dans le registre à décalage du codeur, les  $J$  équations de sommes de contrôles sont calculées de la façon suivante :

$$B_k^j = P_{k+p_j} \cdot \sum_{i=1}^J \begin{cases} \hat{I}_{k+p_j-p_i} & p_j < p_i \\ I_{k+p_j-p_i} & p_j \geq p_i \end{cases} \quad j = 1, 2, \dots, J \quad (2-7)$$

où  $P$  est la parité à l'instant  $(k+p_j)$ ,  $I$  est l'information à l'entrée du décodeur à seuil à l'instant  $(k+p_j-p_i)$  et  $\hat{I}$  est l'information estimée par le décodeur à l'instant  $(k+p_j-p_i)$ . Il est important de noter que la sommation est de type modulo 2 pour un décodeur en quantification dure. Par exemple, avec le code utilisé dans cette étude, la matrice des positions est  $p=(0,7,10,16,18,30,31,35)$  et les équations du décodeur à l'instant  $k=0$  sont :

$$B_0^1 = P_0 \oplus I_0 \oplus \hat{I}_{-7} \oplus \hat{I}_{-10} \oplus \hat{I}_{-16} \oplus \hat{I}_{-30} \oplus \hat{I}_{-31} \oplus \hat{I}_{-35}, \quad (2-8)$$

$$B_0^2 = P_7 \oplus I_7 \oplus I_0 \oplus \hat{I}_{-3} \oplus \hat{I}_{-9} \oplus \hat{I}_{-11} \oplus \hat{I}_{-23} \oplus \hat{I}_{-24} \oplus \hat{I}_{-28}, \quad (2-9)$$

$$B_0^3 = P_{10} \oplus I_{10} \oplus I_3 \oplus I_0 \oplus \hat{I}_{-6} \oplus \hat{I}_{-8} \oplus \hat{I}_{-20} \oplus \hat{I}_{-21} \oplus \hat{I}_{-25}, \quad (2-10)$$

$$B^4_0 = P_{16} \oplus I_{16} \oplus I_9 \oplus I_6 \oplus I_0 \oplus \hat{I}_{-2} \oplus \hat{I}_{-14} \oplus \hat{I}_{-15} \oplus \hat{I}_{-19}, \quad (2-11)$$

$$B^5_0 = P_{18} \oplus I_{18} \oplus I_{11} \oplus I_8 \oplus I_2 \oplus I_0 \oplus \hat{I}_{-12} \oplus \hat{I}_{-13} \oplus \hat{I}_{-17}, \quad (2-12)$$

$$B^6_0 = P_{30} \oplus I_{30} \oplus I_{23} \oplus I_{20} \oplus I_{14} \oplus I_{12} \oplus I_0 \oplus \hat{I}_{-1} \oplus \hat{I}_{-5}, \quad (2-13)$$

$$B^7_0 = P_{31} \oplus I_{31} \oplus I_{24} \oplus I_{21} \oplus I_{14} \oplus I_{13} \oplus I_1 \oplus I_0 \oplus \hat{I}_{-4}, \quad (2-14)$$

$$B^8_0 = P_{35} \oplus I_{35} \oplus I_{28} \oplus I_{25} \oplus I_{19} \oplus I_{17} \oplus I_5 \oplus I_4 \oplus I_0. \quad (2-15)$$

L'estimation de l'information  $\hat{I}_k$  est obtenue grâce à un vote de majorité sur les  $J$  équations de sommes de contrôles ci-dessus. Donc, l'information estimée est le résultat obtenu par  $J/2$  équations ou plus.

En quantification douce, l'information et la parité sont quantifiées et leur niveau représente leur fiabilité. Les équations de sommes de contrôles définies en 2-7 demeurent valides à l'exception que les opérateurs d'addition modulo 2 sont remplacés par des opérateurs appelés «addmin». L'opérateur addmin utilisé dans le cadre de cette thèse est en réalité une approximation de l'opérateur addmin proposé par Lavoie [13]. L'approximation de l'addmin entre deux opérandes est simplement la multiplication des signes de chacun des opérandes et du minimum entre la valeur absolue du premier opérande et celle du deuxième opérande. L'équation 2-16 résume l'opérateur addmin simplifié :

$$\text{Addmin}(E,F) = \text{signe}(E) \bullet \text{signe}(F) \bullet \text{minimum}(|E|, |F|). \quad (2-16)$$

De plus, en quantification douce, l'estimation de l'information est produite par une comparaison de la somme signée des  $J$  équations avec un seuil prédéterminé. Par exemple, pour la quantification définie dans la section précédente, le seuil est zéro. Si la somme est en dessous de zéro le bit d'information est estimé à «1» sinon le bit d'information est estimé à «0».

Les modifications au décodeur, proposées en [10], se basent sur les composantes principales que l'on retrouve dans les équations de sommes de contrôles du décodeur. En examinant attentivement les équations 2-8 à 2-15, on peut voir qu'elles nécessitent :

- ✓ L'information et la parité à l'instant  $k$  (i.e. indice = 0),
- ✓ L'information et la parité à des instants futurs de l'instant  $k$  (i.e. indice > 0),
- ✓ L'information estimée à des instants précédents  $k$  (i.e. indice < 0).

Afin d'intégrer le décodeur à seuil dans la boucle de rétroaction de l'égaliseur à retour d'état, il a été déterminé que l'information et la parité à l'instant  $k$  (i.e. indice = 0) pouvaient provenir de la sommation quantifiée du filtre avant et du filtre arrière. L'ajout d'une ligne à délai à la sortie du filtre avant est par contre nécessaire pour synchroniser l'information et la parité de l'instant  $k$  avec l'information et la parité à des instants futurs et l'information estimée à des instants précédents.

La partie avant du décodeur est constituée des équations qui contiennent l'information et la parité à des instants futurs de  $k$  (i.e. indice > 0). Ces derniers proviennent du filtre à réponse impulsionnelle infinie complexe après quantification. D'autre part, l'information estimée à des instants précédents  $k$  (i.e. indice < 0) est la décision quantifiée du bloc de décision.

Une dernière modification était requise pour pouvoir utiliser le filtre arrière de l'architecture. Le décodeur ne donnant que l'information estimée à sa sortie, il était nécessaire d'ajouter un nouveau calcul de parité basé sur les informations estimées dans le décodeur. Les équations du calcul de la parité sont exactement celles du codeur de l'émetteur à l'exception que l'opérateur addmin est utilisé pour ajouter une fiabilité à la parité estimée. L'équation 2-17 illustre le nouveau calcul de parité :

$$\hat{P}_k = \text{Addmin}(\hat{I}_k, \hat{I}_{k-7}, \hat{I}_{k-10}, \hat{I}_{k-16}, \hat{I}_{k-18}, \hat{I}_{k-30}, \hat{I}_{k-31}, \hat{I}_{k-35}). \quad (2-17)$$

### 2.2.5.1 Performances du décodeur à seuil en quantification douce

Les performances du décodeur à seuil en quantification douce à six niveaux et utilisant le code  $R=1/2$  présenté dans cette section ont été simulées avec le logiciel SPW («Signal Processing Workstation»). La figure 2-6 présente la probabilité d'erreur du décodeur à seuil en quantification douce et la probabilité d'erreur théorique de la modulation QPSK en fonction du rapport d'énergie par bit sur la densité spectrale du bruit ( $E_b/N_0$ ).

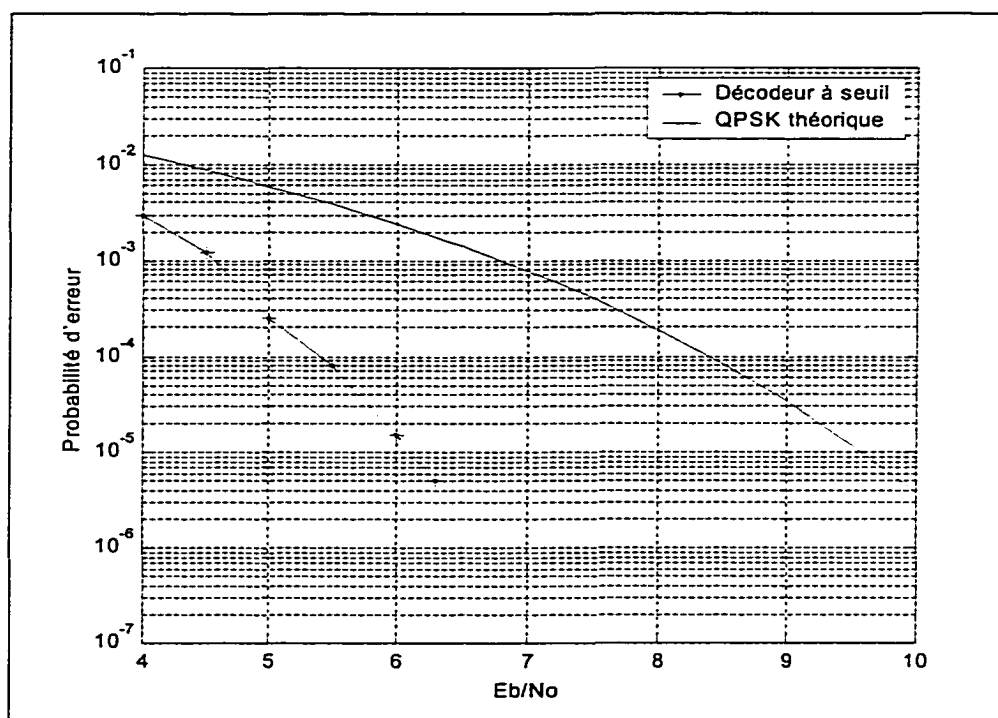


Figure 2-6 Probabilité d'erreur en fonction du rapport  $E_b/N_0$ .

Les simulations ont été faites avec un canal à bruit blanc gaussien additif sans interférence intersymbole. On peut voir que, pour une probabilité d'erreur de  $1 \times 10^{-5}$ , le codage donne un gain de 3.5dB sur la modulation QPSK seulement.

### 2.3 Exemple de performances de l'égaliseur à retour d'état avec décodeur intégré

Pour mieux illustrer l'effet positif de l'intégration d'un décodeur à seuil dans la boucle de rétroaction d'un égaliseur à retour d'état, un résultat de simulation tiré de [10] est présenté avec la figure 2-7. Dans le cadre de ces simulations, le code utilisé était un code  $R=3/4$  et la modulation était de type 16QAM. Il est important de souligner que ce résultat n'est donné qu'à titre d'exemple et qu'il ne servira pas de base de comparaison pour la réalisation de l'architecture.

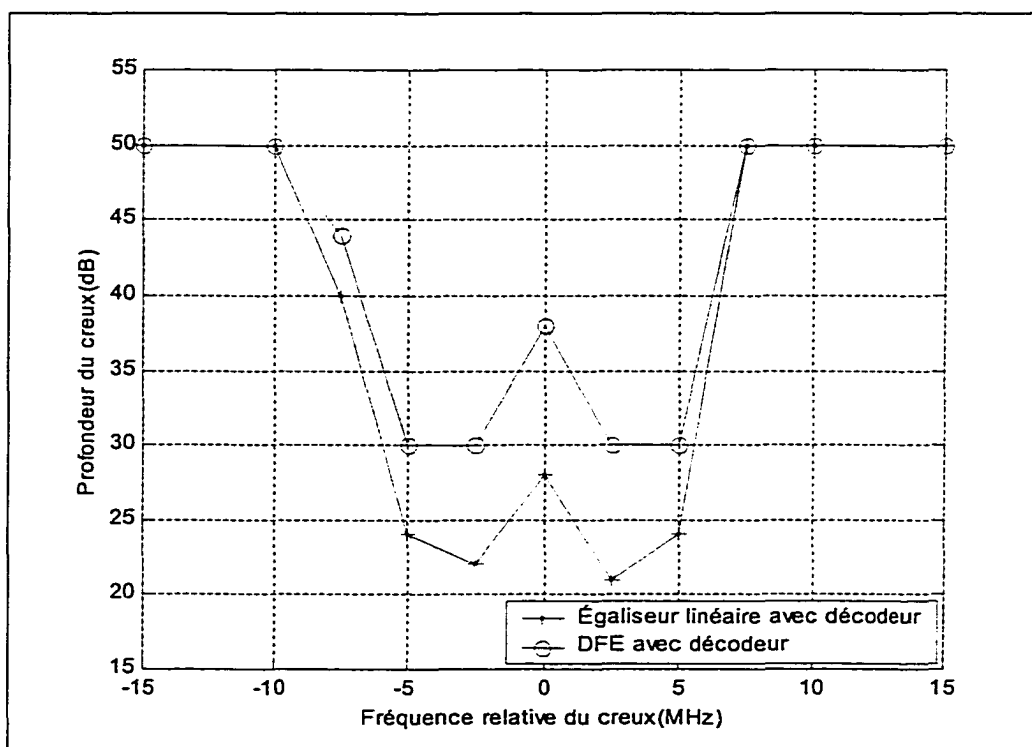


Figure 2-7 Courbes W sans bruit.

La figure 2-7 présente la courbe W (W-curve) pour chacun des types d'égaliseur avec décodeur. Ces courbes nous donnent la sensibilité des égaliseurs aux dispersions multi-chemins. Les régions à l'intérieur des W correspondent aux conditions pour lesquelles le taux d'erreur dépasse le seuil de  $1 \times 10^{-3}$ . Les régions à l'extérieur des W correspondent à

un taux d'erreur inférieur à ce seuil. On voit bien que l'égaliseur à retour d'état avec décodeur est meilleur que l'égaliseur linéaire avec décodeur. En effet, pour une même fréquence de l'évanouissement, le DFE avec décodeur donne des gains de 7 à 10dB.

## 2.4 Conclusion

Ce chapitre a présenté l'architecture de l'égaliseur à retour d'état avec décodeur intégré qui a été réalisée dans ce mémoire. L'architecture est une adaptation pour un code  $R=1/2$  et pour une modulation QPSK de l'architecture proposée par Ladouceur en 1999 [10]. Chacun des composants importants comme le filtre avant, le filtre arrière, le filtre à réponse impulsionnelle infinie complexe, le quantificateur, le modulateur et le décodeur à seuil a été présenté. Nous avons également vu les performances en simulation du décodeur à seuil utilisé dans ce mémoire et un exemple de performances de l'architecture complète simulée en [10] qui explique l'intérêt pour la réalisation de l'égaliseur à retour d'état avec décodeur intégré. Les détails de la réalisation proprement dite seront donnés au chapitre suivant.



## CHAPITRE 3

### RÉALISATION DE L'ÉGALISEUR À RETOUR D'ÉTAT AVEC DÉCODEUR INTÉGRÉ

#### 3.1 Introduction

Ce chapitre décrit comment est réalisé l'égaliseur à retour d'état avec décodeur intégré présenté au chapitre 2. La réalisation matérielle de l'architecture est rendue possible grâce à une radio programmable développée au Laboratoire de communication et d'intégration de la micro-électronique. Cette radio est composée, entre autres choses, de trois circuits programmables FPGA de la compagnie Xilinx. Un de ces circuits est dédié à l'émetteur, un autre à un prédistordeur (qui n'est pas utilisé ici) et le troisième au récepteur. L'égaliseur est réalisé dans le circuit programmable du récepteur. La figure 3-1 présente les principaux circuits qui composent le récepteur.

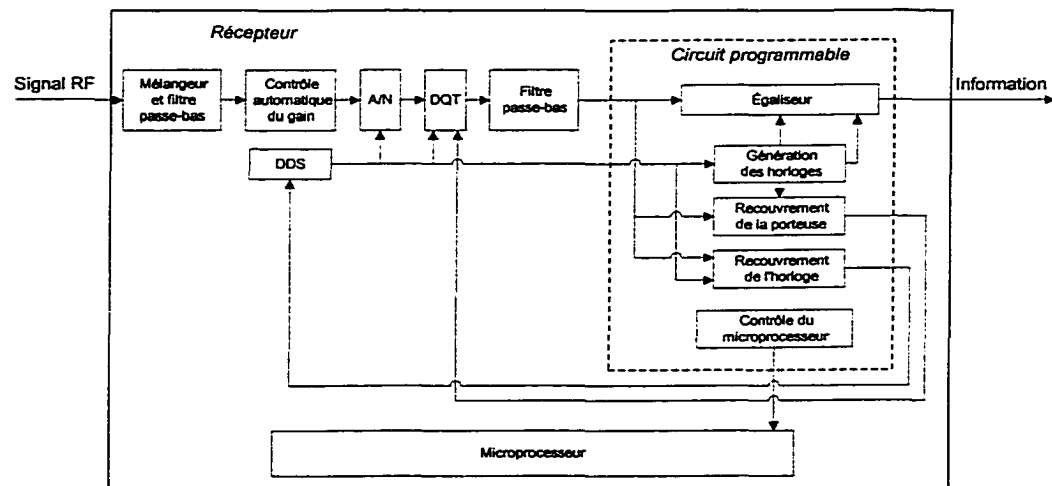


Figure 3-1 Schéma bloc simplifié du récepteur.

Le signal radiofréquence provenant de l'émetteur est mélangé et filtré pour donner un signal situé à la fréquence intermédiaire de 10.976 MHz. Ce signal est amplifié par un circuit de contrôle automatique du gain et converti en un signal numérique avec un convertisseur analogique à numérique (A/N). Ce signal numérique entre dans un DQT («Digital Quadrature Tuner») qui permet d'extraire les composantes en quadrature du signal (i.e. les composantes I et Q). Ces dernières sont filtrées avant d'entrer dans le circuit programmable. D'autre part, le DDS est un circuit générateur d'horloge programmable utilisé pour produire une horloge de référence avec laquelle les horloges nécessaires au récepteur seront dérivées. Enfin, le microprocesseur permet la programmation et le contrôle des différents circuits intégrés de la radio programmable.

### **3.1.1 Le récepteur**

Le circuit programmable du récepteur est de type XLA4085. Cinq (5) blocs sont principalement réalisés à l'intérieur de ce circuit : l'égaliseur à retour d'état avec décodeur intégré, le générateur des horloges, le circuit de recouvrement de la porteuse, le circuit de recouvrement de l'horloge et le circuit de contrôle du microprocesseur. Les blocs autour de l'égaliseur ont été conçus au LACIME dans le cadre de projets précédents. Tous les blocs incluant l'égaliseur sont réalisés d'abord sous forme de portes logiques avec le logiciel Powerview de la compagnie Viewlogic Systems avant d'être traduit dans un format binaire de programmation avec le compilateur de Xilinx.

#### **3.1.1.1 Le circuit programmable XLA4085**

Le circuit programmable XLA4085 est constitué de cellules logiques programmables ou CLB («Configurable Logic Blocks») qui possèdent [19] :

- ✓ Treize (13) entrées et quatre (4) sorties,
- ✓ Trois (3) générateurs de fonction et deux (2) bascules D.

Le compilateur a pour rôle de réaliser les fonctions conçues avec des portes logiques dans les cellules logiques programmables. Chaque cellule permet de créer la ou les fonctions logiques suivantes :

- ✓ N'importe quelle fonction d'au plus quatre (4) variables d'entrée. Une deuxième fonction d'au plus quatre (4) variables d'entrée qui ne sont pas communes à la première fonction. Une troisième fonction d'au plus trois (3) variables d'entrée qui ne sont pas communes à la première et à la deuxième fonction. La sortie de cette dernière fonction doit cependant passer par une bascule D.
- ✓ N'importe quelle fonction de cinq (5) variables d'entrée.
- ✓ N'importe quelle fonction de quatre (4) variables d'entrée et une deuxième fonction de six (6) variables d'entrée avec quelques variables communes à la première fonction.
- ✓ Une fonction d'au plus neuf (9) variables d'entrée.

La réalisation de fonctions plus complexes est possible avec une combinaison de plusieurs cellules logiques programmables. De plus, les cellules logiques peuvent être utilisées comme de la mémoire vive de 16x2 bits à un port, 32x1 bit à un port ou 16x1 bit à deux ports.

Il est possible d'utiliser 3136 cellules logiques programmables à l'intérieur du circuit XLA4085 du récepteur. Cette limite est une des contraintes importantes de ce mémoire. En effet, les quatre (4) autres blocs du circuit programmable utilisent déjà 600 CLB. Il reste donc 2536 cellules logiques programmables disponibles pour réaliser et déverminer l'égaliseur à retour d'état avec décodeur intégré.

Une seconde contrainte importante est le temps de propagation nécessaire au signal pour passer d'une cellule logique à une autre. En général, ce temps de propagation ne doit jamais dépasser la période des différentes horloges présentes dans le circuit. Par

exemple, un circuit complexe requiert un nombre élevé de cellules logiques et il est possible que le temps de propagation entre les cellules soit plus grand que la période de l'horloge. Les solutions à ce problème sont, entre autres, de choisir des fréquences d'horloge plus basses et d'utiliser la technique du pipeline lorsque c'est possible.

### 3.1.1.2 Les horloges

La figure 3-2 présente le schéma bloc de l'égaliseur à retour d'état avec décodeur intégré. La quantification de chacun des composants est donnée aussi sur cette figure.

Trois (3) horloges sont utilisées dans l'égaliseur. La première horloge (rx\_baud\_clk) a une fréquence égale au taux de symbole. La deuxième horloge (rx\_2xbaud\_clk) a une fréquence égale au double du taux de symbole. La troisième horloge (syst\_clk), qui ne figure pas sur le schéma bloc, est requise pour la programmation des registres et pour le contrôle de l'égaliseur. Toutes ces horloges sont générées à partir d'une horloge maître (M\_clk). La fréquence de ces horloges est donnée dans le tableau 3-1.

Tableau 3-1

Fréquence des horloges de l'égaliseur.

Horloge	Fréquence
Rx_baux_clk	4.096 MHz
Rx_2xbaux_clk	8.192 MHz
Syst_clk	49.152 MHz
M_clk	49.152 MHz

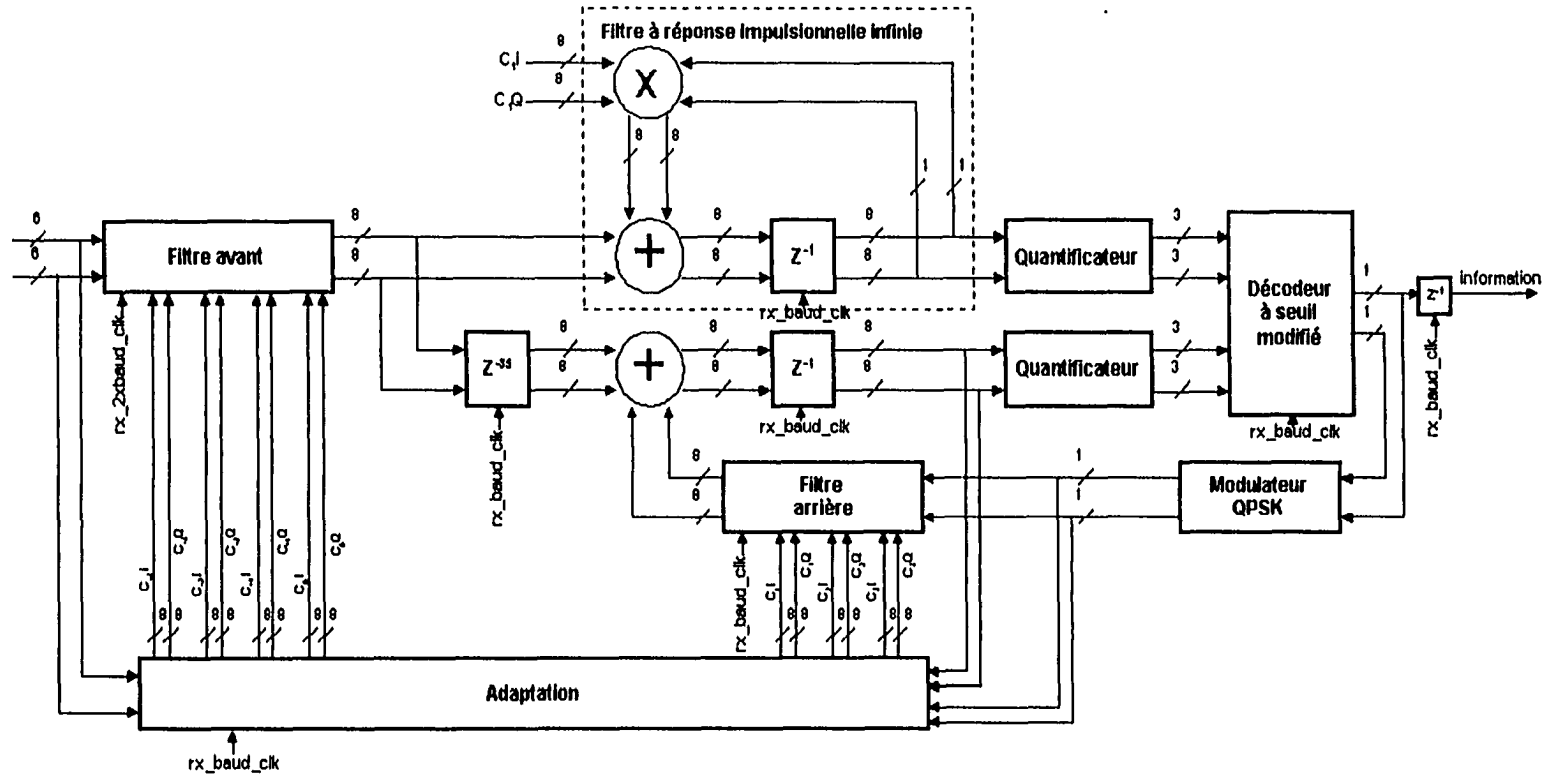


Figure 3-2 Schéma bloc de la réalisation de l'égaliseur à retour d'état avec décodeur intégré

Les sections suivantes expliqueront plus en détail la réalisation des composants de l'égaliseur. Tous les schémas de réalisation en portes logiques avec le logiciel Powerview sont fournis dans l'annexe A. Avant de commencer, il est important de noter que la notation complément à deux est utilisée pour tous les composants. Par contre, la quantification des bits est différente selon le composant. La notation par somme de puissances de deux servira à exprimer la quantification des bits. Par exemple, si les bits «1011» sont sur une entrée de quatre (4) bits (quantification :  $-2^2+2^1+2^0+2^{-1}$ ) cela signifie que la valeur de l'entrée est  $-2.5$  (i.e.  $-2^2+2^0+2^{-1}$ ).

### 3.2 Le filtre avant

Le filtre avant utilise les symboles du DQT qui sont quantifiés avec six (6) bits par dimension (quantification :  $-2^2+2^1+2^0+2^{-1}+2^{-2}+2^{-3}$ ). La moyenne, par dimension, de la valeur absolue des symboles est programmée à 1. Le filtre avant fonctionne au double du taux de symbole soit 8.192MHz.

Les quatre (4) coefficients complexes ( $c_{-3}$ ,  $c_{-2}$ ,  $c_{-1}$  et  $c_0$ ) sont quantifiés avec huit (8) bits par dimension (quantification :  $-2^1+2^0+2^{-1}+2^{-2}+2^{-3}+2^{-4}+2^{-5}+2^{-6}$ ). Cette quantification est choisie en fonction des résultats de simulations avec le logiciel SPW d'un égaliseur semblable en adaptation. Les intervalles de valeurs des coefficients complexes en simulation sont donnés dans le tableau 3-2.

Tableau 3-2

Intervalle de valeurs des coefficients complexes du filtre avant en simulation.

Coefficient	Partie réelle	Partie imaginaire
$c_0$	[0.5 , 1.2]	[-0.7 , 0.7]
$c_{-1}$	[-0.5 , 0.5]	[-0.5 , 0.5]
$c_{-2}$	[-0.5 , 0.5]	[-0.5 , 0.5]
$c_{-3}$	[-0.25 , 0.25]	[-0.25 , 0.25]

Tel que précédemment montré à la figure 2-2, le filtre avant est constitué de trois (3) unités de délai complexe, de quatre (4) multiplicateurs complexes et de trois (3) additionneurs complexes. De façon rigoureuse, la multiplication d'un nombre complexe de six (6) bits par dimension (quantification :  $-2^2+2^1+2^0+2^{-1}+2^{-2}+2^{-3}$ ) avec un autre nombre complexe de huit (8) bits par dimension (quantification :  $-2^1+2^0+2^{-1}+2^{-2}+2^{-3}+2^{-4}+2^{-5}+2^{-6}$ ) donne un résultat complexe de quinze (15) bits par dimension (quantification :  $-2^5+2^4+2^3+2^2+2^1+2^0+2^{-1}+2^{-2}+2^{-3}+2^{-4}+2^{-5}+2^{-6}+2^{-7}+2^{-8}+2^{-9}$ ). D'autre part, l'addition de deux nombres complexes ajoute un bit dans la partie la plus significative du résultat de l'addition. Ainsi, la somme des quatre (4) multiplications complexes nécessite dix-huit (18) bits par dimension (quantification :  $-2^8+2^7+2^6+2^5+2^4+2^3+2^2+2^1+2^0+2^{-1}+2^{-2}+2^{-3}+2^{-4}+2^{-5}+2^{-6}+2^{-7}+2^{-8}+2^{-9}$ ).

La contrainte sur le nombre de cellules logiques programmables impose de déterminer quels bits sont réellement nécessaires lors d'une opération normale de l'égaliseur. Tout d'abord, les bits qui ont une quantification plus petite que  $2^{-3}$  sont inutiles puisque l'entrée du filtre avant n'est pas plus petite que  $2^{-3}$ . De plus, les intervalles de valeurs des coefficients présentées dans le tableau 3-2 nous permettent de croire que les deux bits les plus significatifs de la multiplication complexe d'un symbole par n'importe quel des coefficients ne sont pas nécessaires. L'ajout d'un bit à chaque addition est également superflu si on considère que :

- ✓ Les symboles à l'entrée du filtre sont aléatoires,
- ✓ Il y a un nombre égal de valeurs positives et de valeurs négatives et ce autant pour les symboles à l'entrée que pour les coefficients du filtre,
- ✓ La valeur des parties réelle et imaginaire des coefficients autres que le coefficient central est plus petite que 1.

Ainsi, en tenant compte de ces hypothèses, la sortie du filtre est fixée à huit (8) bits par dimension (quantification :  $-2^3+2^2+2^1+2^0+2^{-1}+2^{-2}+2^{-3}+2^{-4}$ ).

Les schémas dans l'annexe A présente la réalisation du filtre avant. Il est composé de six (6) bascules D de six (6) bits qui forment trois (3) unités de délai complexe, de quatre (4) multiplicateurs complexes 6x8 bits et de six (6) additionneurs de huit (8) bits qui sont, en fait, trois (3) additionneurs complexes de huit (8) bits. Des détecteurs de dépassement sont aussi présents pour vérifier que le filtre avant fonctionne normalement.

Le multiplicateur complexe 6x8 bits est réalisé à l'aide de quatre (4) multiplicateurs 6x8 bits scalaires générés par le logiciel CoreGen de Xilinx, d'un additionneur et d'un soustracteur de huit (8) bits. L'avantage des multiplicateurs du logiciel CoreGen est qu'ils sont générés de la façon à utiliser le moins de cellules logiques possible et de pouvoir fonctionner à des fréquences élevées. Ils possèdent, par contre, une latence de quatre (4) coups d'horloge pour produire le résultat d'une multiplication. L'agencement des multiplicateurs scalaires, de l'additionneur et du soustracteur est tel que :

$$\text{Re}\{O\} = \text{Re}\{A\} \cdot \text{Re}\{C\} - \text{Im}\{A\} \cdot \text{Im}\{C\}, \quad (3-1)$$

$$\text{Im}\{O\} = \text{Im}\{A\} \cdot \text{Re}\{C\} + \text{Re}\{A\} \cdot \text{Im}\{C\}, \quad (3-2)$$

où O est le résultat de la multiplication du symbole A et du coefficient C.

Des bascules de huit (8) bits ont été ajoutées à la sortie de chaque multiplicateur complexe et à la sortie du filtre avant. Ce pipeline est nécessaire à cause de la complexité du filtre avant et de sa fréquence de fonctionnement. En conséquence, la latence du filtre est neuf (9) coups d'horloge à deux fois le taux de symbole. La réalisation du filtre avant nécessite l'utilisation de 780 cellules logiques programmables.

En terminant cette section, il est important de noter que les composants suivants le filtre avant utilisent la sortie de ce dernier mais à une fois le taux de symbole. Un bloc de choix de symboles est alors nécessaire pour présenter le bon symbole aux autres



composants (voir le schéma dans l'annexe A). La latence combinée du filtre avant et du bloc de choix de symboles est alors de six (6) coups d'horloge à une fois le taux de symbole.

### 3.3 Le filtre arrière

Le filtre arrière utilise l'information et la parité remodulées et quantifiées avec 1 bit par dimension (voir la section 3.6 sur le modulateur). Ce filtre fonctionne au taux de symbole soit 4.096MHz.

Les trois (3) coefficients complexes ( $c_1$ ,  $c_2$  et  $c_3$ ) sont quantifiés avec huit (8) bits par dimension (quantification :  $-2^0+2^{-1}+2^{-2}+2^{-3}+2^{-4}+2^{-5}+2^{-6}+2^{-7}$ ). Cette quantification est déterminée aussi avec les simulations sur SPW. Les intervalles de valeurs des coefficients complexes sont présentés dans le tableau 3-3.

Tableau 3-3

Intervalle de valeurs des coefficients complexes du filtre arrière en simulation.

Coefficient	Partie réelle	Partie imaginaire
$c_1$	$[-0.5, 0.5]$	$[-0.5, 0.5]$
$c_2$	$[-0.25, 0.25]$	$[-0.25, 0.25]$
$c_3$	$[-0.10, 0.10]$	$[-0.10, 0.10]$

Comme pour le filtre avant, les intervalles de valeurs des coefficients ont permis de déterminer les bits nécessaires. La sortie du filtre est ainsi fixée à huit (8) bits par dimension (quantification :  $-2^1+2^0+2^{-1}+2^{-2}+2^{-3}+2^{-4}+2^{-5}+2^{-6}$ ).

Tel que montré précédemment à la figure 2-3, le filtre arrière est constitué de trois (3) unités de délai complexe, de trois (3) multiplicateurs complexes et de deux (2)

additionneurs complexes. La première unité de délai est déplacée à l'extérieur du filtre arrière pour être placée après la somme des symboles du filtre avant et du filtre arrière, comme présenté à la figure 3-2. Ce déplacement ne change pas l'équation du filtre arrière et permet de couper en deux la chaîne de rétroaction qui débute à la sortie du décodeur et se termine dans le bloc de décision du décodeur (voir la figure 2-1).

Le filtre arrière est donc réalisé avec deux (2) bascules D d'un (1) bit, trois (3) multiplicateurs complexes 8x1 bits et quatre (4) additionneurs de huit (8) bits qui forment deux (2) additionneurs complexes de huit (8) bits. Les schémas de réalisation sont à l'annexe A.

Le multiplicateur complexe 8x1 bits est particulier car il est réalisé à l'aide de quatre (4) blocs de complément à deux du coefficient et de deux additionneurs de huit (8) bits. Les blocs de complément à deux sont commandés par le symbole à l'entrée du filtre arrière. Ceci est possible à cause de la signification du bit de chaque dimension. Un «0» représente la valeur 1 et un «1» représente la valeur -1. L'opération du multiplicateur complexe 8x1 bits peut se résumer aussi avec les équations 3-1 et 3-2 mais avec  $\text{Re}\{A\} = \pm 1$  et  $\text{Im}\{A\} = \pm 1$  selon le symbole.

La technique du pipeline n'est pas permise dans le filtre arrière en raison de sa présence dans la boucle de rétroaction de l'égaliseur à retour d'état. Sa latence est donc nulle. La réalisation du filtre arrière requiert cent (100) cellules logiques programmables.

### 3.4 Le filtre à réponse impulsionnelle infinie complexe

Le filtre à réponse impulsionnelle infinie complexe utilise les symboles du filtre avant quantifiés avec huit (8) bits par dimension ( $-2^3+2^2+2^1+2^0+2^{-1}+2^{-2}+2^{-3}+2^{-4}$ ). Ce filtre fonctionne au taux de symbole. Le coefficient complexe ( $c_1$ ) est quantifié avec huit (8) bits par dimension (quantification :  $-2^0+2^{-1}+2^{-2}+2^{-3}+2^{-4}+2^{-5}+2^{-6}+2^{-7}$ ).

Tel que précédemment montré à la figure 2-4, le filtre RII est composé d'une unité de délai complexe, d'un multiplicateur complexe et d'un additionneur complexe. L'unité de délai est déplacée à la sortie de l'additionneur complexe comme le montre la figure 3-2. Ce déplacement ne change en rien l'équation du filtre et permet de compenser le délai du filtre arrière présent dans la chaîne de rétroaction de l'égaliseur.

Seul le signe du symbole à la sortie du filtre est utilisé dans la multiplication avec le coefficient et ce pour réduire la complexité du filtre. De plus, le bit le plus significatif du résultat de l'addition complexe est jugé inutile et ce en supposant que la valeur des symboles à la sortie du filtre RII doit se rapprocher de la valeur des symboles transmis à l'émetteur (i.e.  $\pm 1$  pour les parties réelle et imaginaire). La sortie du filtre RII est fixée à huit (8) bits par dimension (quantification :  $-2^2+2^1+2^0+2^{-1}+2^{-2}+2^{-3}+2^{-4}+2^{-5}$ ).

Ainsi, le filtre complexe à réponse impulsionnelle infinie est réalisé avec un multiplicateur 8x1 bits (le même multiplicateur qu'au filtre arrière), deux additionneurs de huit (8) bits qui forment l'additionneur complexe et deux bascules D de huit (8) bits. Un détecteur de dépassement est aussi présent. La latence du filtre est un coup d'horloge et sa complexité est d'environ quarante (40) cellules logiques programmables.

### 3.5 Le quantificateur

Le quantificateur réalisé ici est celui expliqué dans la section 2.2.4 mais avec des modifications pour faciliter la réalisation. Le quantificateur utilise des symboles quantifiés avec 8 bits par dimension (quantification :  $-2^2+2^1+2^0+2^{-1}+2^{-2}+2^{-3}+2^{-4}+2^{-5}$ ).

Pour ne pas trop augmenter la complexité du décodeur à seuil en quantification douce présenté dans la section suivante, la sortie du quantificateur a trois (3) bits qui représentent l'information et trois (3) bits qui représentent la parité (quantification :  $-2^2+2^1+2^0$ ).

Avec trois (3) bits pour quantifier l'information ou pour la parité, il est possible en théorie d'avoir huit (8) poids de fiabilité. Dans le cadre de ce mémoire, six (6) poids sont utilisés afin de respecter le format complément à deux. En effet, le poids 0 «000» n'ajoutent aucune valeur ni dans un sens ni dans l'autre lors d'une addition signée en complément à deux. Ce poids a donc été rejeté puisque le bloc de décision du décodeur se base sur une addition signée des sommes de contrôles. Par le fait même, pour avoir le même nombre de poids positifs et négatifs, le poids le plus négatif -4 «100» a été retiré.

Le tableau 3-4 présente la quantification de l'information selon la valeur de la partie réelle du symbole. Il faut noter que la même quantification est utilisée pour la parité qui se trouve sur la partie imaginaire du symbole.

Tableau 3-4

Quantification de l'information selon l'intervalle sur l'axe réel.

Intervalle sur la l'axe réel	Signification	Sortie du quantificateur
$-\infty, -0.78125[$	Bit d'information est 1 très fiable	-3
$[-0.78125, -0.375[$	Bit d'information est 1 fiable	-2
$[-0.375, 0[$	Bit d'information est 1 peu fiable	-1
$[0, 0.375]$	Bit d'information est 0 peu fiable	1
$]0.375, 0.78125]$	Bit d'information est 0 fiable	2
$]0.78125, \infty$	Bit d'information est 0 très fiable	3

Les valeurs entières (i.e. -3, -2, -1, 1, 2, 3) servent de poids à la place des poids proposés à la figure 2-6 et ce pour faciliter la réalisation de l'opérateur addmin utilisé dans le décodeur. Les régions de quantification, par contre, sont à peu près les mêmes en tenant compte de la quantification des symboles à l'entrée du quantificateur.

Le schéma bloc de la quantification de l'information réalisée dans le cadre de ce mémoire est présenté à la figure 3-3. Le même traitement est fait pour la quantification de la parité. Les schémas de réalisation sous forme de portes logiques sont donnés dans l'annexe A.

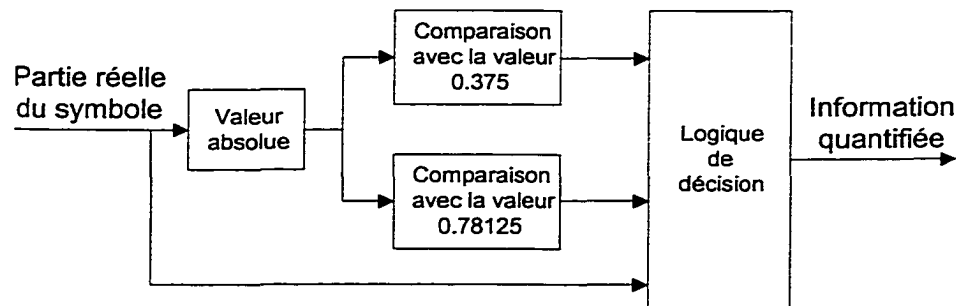


Figure 3-3 Schéma bloc du quantificateur de l'information.

Le quantificateur est réalisé avec des comparateurs qui comparent la valeur absolue de l'entrée avec les limites des régions de quantification et qui donnent l'intervalle où se situe l'entrée à une logique de décision. Le signe de l'entrée est utilisé aussi dans la logique de décision pour déterminer le bon poids. Les comparateurs s'avèrent pratiques pour changer les régions de quantification.

Enfin, le quantificateur a une latence nulle puisqu'un des quantificateurs se situe dans la boucle de rétroaction de l'égaliseur à retour d'état. Une quarantaine de cellules logiques programmables sont utilisées pour sa réalisation.

### 3.6 Le décodeur à seuil à quantification douce modifié

Nous avons vu au chapitre 2 que le décodeur à seuil modifié est caractérisé par les équations 2-8 à 2-15 et par l'équation 2-17. L'analyse de ces équations permet de déterminer que la réalisation du décodeur exige :

- ✓ Trente-cinq (35) unités de délai pour l'information à des instants futurs de  $k$  (i.e. indice  $> 0$ ),
- ✓ Trente-cinq (35) unités de délai pour la parité à des instants futurs de  $k$  (i.e. indice  $> 0$ ),
- ✓ Trente-cinq (35) unités de délai pour l'information estimée à des instants précédents  $k$  (i.e. indice  $< 0$ ),
- ✓ Soixante-trois (63) opérateurs addmin à deux opérandes.

Pour les besoins de la réalisation (voir les schémas à l'annexe A), la partie avant du décodeur en quantification douce (trois (3) bits de quantification) est constituée des équations qui contiennent l'information à des instants futurs de  $k$  (i.e. indice  $> 0$ ). Cette information provient du filtre à réponse impulsionnelle infinie complexe. Les équations de la partie avant à l'instant  $k=0$  sont donc :

$$B^{2A}_0 = I_7, \quad (3-3)$$

$$B^{3A}_0 = I_{10} \wedge I_3, \quad (3-4)$$

$$B^{4A}_0 = I_{16} \wedge I_9 \wedge I_6, \quad (3-5)$$

$$B^{5A}_0 = I_{18} \wedge I_{11} \wedge I_8 \wedge I_2, \quad (3-6)$$

$$B^{6A}_0 = I_{30} \wedge I_{23} \wedge I_{20} \wedge I_{14} \wedge I_{12}, \quad (3-7)$$

$$B^{7A}_0 = I_{31} \wedge I_{24} \wedge I_{21} \wedge I_{14} \wedge I_{13} \wedge I_1, \quad (3-8)$$

$$B^{8A}_0 = I_{35} \wedge I_{28} \wedge I_{25} \wedge I_{19} \wedge I_{17} \wedge I_5 \wedge I_4, \quad (3-9)$$

où  $\wedge$  représente l'opérateur addmin à deux entrées.

La partie avant demande donc trente-cinq (35) bascules D de trois (3) bits et vingt-et-un (21) opérateurs addmin.

Une autre partie du décodeur permet de compléter ces équations avec la parité à des instants futurs de  $k$  (i.e. indice  $> 0$ ) et l'information estimée à des instants précédents  $k$  (i.e. indice  $< 0$ ). La parité provient aussi du filtre à réponse impulsionnelle infinie. Cette partie permet également de calculer la nouvelle parité NP nécessaire à l'intégration du décodeur dans la boucle de rétroaction de l'égaliseur. Les équations de cette partie à l'instant  $k=0$  sont :

$$B^{1B}_0 = \hat{I}_{-7} \wedge \hat{I}_{-10} \wedge \hat{I}_{-16} \wedge \hat{I}_{-30} \wedge \hat{I}_{-31} \wedge \hat{I}_{-35}, \quad (3-10)$$

$$B^{2B}_0 = P_7 \wedge \hat{I}_{-3} \wedge \hat{I}_{-9} \wedge \hat{I}_{-11} \wedge \hat{I}_{-23} \wedge \hat{I}_{-24} \wedge \hat{I}_{-28}, \quad (3-11)$$

$$B^{3B}_0 = P_{10} \wedge \hat{I}_{-6} \wedge \hat{I}_{-8} \wedge \hat{I}_{-20} \wedge \hat{I}_{-21} \wedge \hat{I}_{-25}, \quad (3-12)$$

$$B^{4B}_0 = P_{16} \wedge \hat{I}_{-2} \wedge \hat{I}_{-14} \wedge \hat{I}_{-15} \wedge \hat{I}_{-19}, \quad (3-13)$$

$$B^{5B}_0 = P_{18} \wedge \hat{I}_{-12} \wedge \hat{I}_{-13} \wedge \hat{I}_{-17}, \quad (3-14)$$

$$B^{6B}_0 = P_{30} \wedge \hat{I}_{-1} \wedge \hat{I}_{-5}, \quad (3-15)$$

$$B^{7B}_0 = P_{31} \wedge \hat{I}_{-4}, \quad (3-16)$$

$$B^{8B}_0 = P_{35}. \quad (3-17)$$

$$NP_0 = I_0 \wedge \hat{I}_{-7} \wedge \hat{I}_{-10} \wedge \hat{I}_{-16} \wedge \hat{I}_{-30} \wedge \hat{I}_{-31} \wedge \hat{I}_{-35}. \quad (3-18)$$

Il est possible de réduire le nombre d'opérateurs addmin pour cette partie du décodeur. Pour ce faire, il s'agit d'insérer des calculs sur l'information estimée directement dans le registre à décalage de l'information estimée comme le montre la figure 3-4. Cette simplification permet d'utiliser treize (13) opérateurs addmin au lieu de trente-deux (32). La réalisation de cette partie du décodeur exige donc soixante-dix (70) bascules D de trois (3) bits en plus des treize (13) opérateurs addmin.





En supposant que les XXX représentent un état indifférent, les équations booléennes des trois (3) bits de la sortie O sont :

$$O_2 = A_2 \oplus B_2 , \quad (3-19)$$

$$O_1 = A_2 \overline{B_2} \overline{B_1} + A_2 \overline{B_2} B_1 + \overline{A_2} B_2 B_1 + \overline{A_1} B_2 \overline{B_1} \\ + A_1 \overline{A_0} B_1 \overline{B_0} + A_1 \overline{A_0} B_2 \overline{B_1} + A_2 \overline{A_1} B_1 \overline{B_0} , \quad (3-20)$$

$$O_0 = \overline{B_2} \overline{B_1} + \overline{A_2} \overline{A_1} + A_2 A_1 A_0 + B_2 B_1 B_0 + A_0 B_0 . \quad (3-21)$$

Le nombre de cellules logiques programmables requis pour un opérateur addmin à deux opérandes est sept (7) cellules.

Avant d'entrer dans le bloc de décision du décodeur, les deux parties de chacune des équations 3-3 à 3-18 sont regroupées avec une opération addmin pour former les équations  $B^1$  à  $B^8$  :

$$B^1_0 = B^{1B}_0 \wedge P_0 , \quad (3-22)$$

$$B^2_0 = B^{2A}_0 \wedge B^{2B}_0 , \quad (3-23)$$

$$B^3_0 = B^{3A}_0 \wedge B^{3B}_0 , \quad (3-24)$$

$$B^4_0 = B^{4A}_0 \wedge B^{4B}_0 , \quad (3-25)$$

$$B^5_0 = B^{5A}_0 \wedge B^{5B}_0 , \quad (3-26)$$

$$B^6_0 = B^{6A}_0 \wedge B^{6B}_0 , \quad (3-27)$$

$$B^7_0 = B^{7A}_0 \wedge B^{7B}_0 , \quad (3-28)$$

$$B^8_0 = B^{8A}_0 \wedge B^{8B}_0 . \quad (3-29)$$

Il est important de rappeler que la parité à l'instant  $k=0$  (i.e.  $P_0$ ) provient de la somme quantifiée des symboles du filtre avant et du filtre arrière.

Le dernier composant du décodeur à seuil est le bloc de décision. Ce bloc fait la somme signée des équations  $B^1$  à  $B^8$  et de l'information à l'instant  $k=0$  (i.e.  $I_0$ ). Cette dernière provient aussi de la somme quantifiée des symboles du filtre avant et du filtre arrière. Six (6) bits sont nécessaires pour la somme signée afin de couvrir toute la plage possible de  $-27$  à  $27$  (i.e.  $-3 \times 9$  équations à  $+3 \times 9$  équations). Cette somme est ensuite limitée aux valeurs admissibles de l'opérateur addmin (i.e.  $-3$  à  $3$ ) pour pouvoir être retournée au registre à décalage de l'information estimée.

Une protection supplémentaire est ajoutée dans le bloc de décision pour le cas où le résultat de la sommation serait zéro. En effet, cette valeur n'est pas acceptée. Dans ce cas, la valeur de l'information quantifiée à l'instant  $k=0$  (i.e.  $I_0$ ) remplace le zéro. Cette valeur fut choisie comme valeur aléatoire et ce dans le but de ne pas biaiser les décisions.

Enfin, l'information est estimée en comparant le résultat de la somme des neuf (9) équations avec le seuil de décision qui est zéro. Si la somme est plus grande que zéro (i.e. le bit le plus significatif de la somme est un «0»), l'information est estimée comme un «0» et si la somme est plus petite que zéro (i.e. le bit le plus significatif de la somme est un «1»), l'information est estimée comme un «1».

Le décodeur possède une latence équivalente à sa longueur de contrainte soit trente-cinq (35) coups d'horloge. Le nombre de cellules logiques programmables requises pour la réalisation du décodeur est 360 CLB.

### 3.7 Le modulateur

Le modulateur utilise deux bits (l'information et la parité estimées) à son entrée pour produire un symbole d'un bit par dimension.

La réalisation du modulateur est grandement simplifiée à cause de la représentation quadratique de l'information et de la parité présentée à la section 2.2.3. En effet, le tableau 2-1 indique que :

- ✓ Un «0» est associé à la valeur 1,
- ✓ Un «1» est associé à la valeur  $-1$ ,
- ✓ L'information est située sur la composante I (partie réelle du symbole),
- ✓ La parité est située sur la composante Q (partie imaginaire du symbole).

Ainsi, il s'agit de changer la signification des bits à la sortie du modulateur. Si le bit sur l'une ou l'autre des dimensions est un «0» cela veut dire qu'il représente la valeur 1 et si le bit est un «1», il représente la valeur  $-1$ . Le modulateur est donc réalisé avec seulement une cellule logique et a une latence nulle.

### **3.8 La ligne à délai complexe de trente-cinq (35) coups d'horloge**

La ligne à délai complexe de trente-cinq (35) coups d'horloge (huit (8) bits par dimension) fonctionne au taux de symbole. Elle est réalisée avec quatre (4) mémoires 16x8 bits à deux (2) ports, un compteur de quatre (4) bits, un additionneur de quatre (4) bits et dix (10) bascules D de huit (8) bits.

Le contrôle adéquat du port d'écriture et de lecture de deux (2) mémoires 16x8 bits connectées en série permet de créer un délai de trente (30) coups d'horloge (voir le schéma de réalisation à l'annexe A). Cinq (5) bascules D de huit (8) bits sont ajoutées en série pour obtenir les trente-cinq (35) délais. Cette façon de faire permet la réalisation de la ligne à délai complexe avec quatre-vingt (80) cellules logiques programmables. La même réalisation avec uniquement des bascules D demanderait 280 cellules logiques. Une économie donc de 200 CLB.

### 3.9 La somme complexe

La somme complexe des symboles du filtre avant et des symboles du filtre arrière est quantifiée avec huit (8) bits par dimension (quantification :  $-2^2+2^1+2^0+2^{-1}+2^{-2}+2^{-3}+2^{-4}+2^{-5}$ ). Cette quantification de la somme est possible en supposant, comme pour le filtre RII complexe, que la valeur des symboles à la sortie de la somme doit se rapprocher de la valeur des symboles transmis à l'émetteur (i.e.  $\pm 1$  pour les parties réelle et imaginaire). La somme complexe est donc réalisée avec deux (2) additionneurs de huit (8) bits et requiert huit (8) cellules logiques programmables.

### 3.10 L'adaptation

Le bloc d'adaptation est certes le bloc le plus important et le plus complexe de ce mémoire. Il calcule les sept (7) coefficients complexes de l'égaliseur. L'adaptation des coefficients est faite au taux de symbole. La figure 3-5 présente le schéma bloc de la réalisation de l'adaptation.

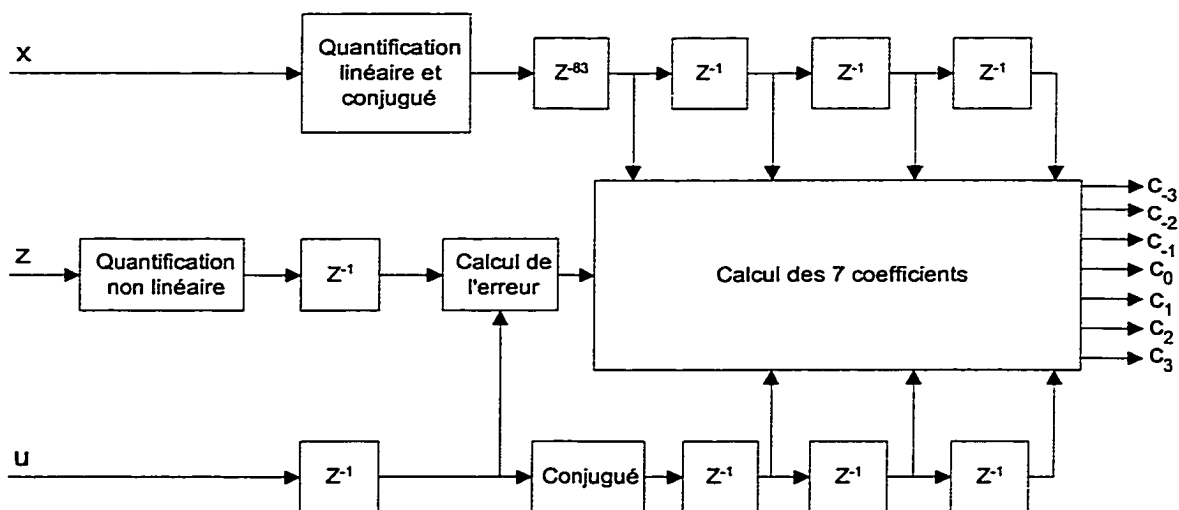


Figure 3-5 Schéma bloc de la réalisation de l'adaptation.

La notation utilisée dans la figure 3-5 respecte la notation utilisée dans la figure 2-1. Ainsi,  $x$  est le symbole à l'entrée de l'égaliseur quantifié sur cinq (5) bits par dimension (quantification :  $-2^2+2^1+2^0+2^{-1}+2^{-2}$ ),  $z$  est le résultat de la somme des symboles du filtre avant et du filtre arrière quantifié sur cinq (5) bits par dimension (quantification :  $-2^2+2^1+2^0+2^{-1}+2^{-2}$ ) et enfin,  $u$  est le symbole quantifié sur 1 bit par dimension résultant de la remodulation de l'information et de la parité estimées.

Les délais sur les symboles avant le calcul des coefficients sont essentiels pour que les coefficients du filtre avant et ceux du filtre arrière s'adaptent en fonction de la même erreur. En analysant attentivement les latences des symboles  $z$  et  $u$  avant d'entrer dans le calcul de l'erreur, il est possible de déterminer le retard nécessaire pour les symboles  $x$ .

Premièrement, la latence des symboles  $z$  est quarante-trois (43) coups d'horloge au taux de symbole. Cette latence est déterminée en additionnant :

- ✓ La latence du filtre avant et du bloc de choix de symboles (i.e. six (6) coups d'horloge),
- ✓ La ligne à délai (i.e. trente-cinq (35) coups d'horloge),
- ✓ La latence du filtre arrière (i.e. un coup d'horloge).
- ✓ Le délai d'un coup d'horloge à la sortie du quantificateur non linéaire.

Ensuite, la latence des symboles  $u$  est aussi quarante-trois (43) coups d'horloge au taux de symbole. Cette latence est déterminée en additionnant :

- ✓ La latence du filtre avant et du bloc de choix de symbole (i.e. six (6) coups d'horloge),
- ✓ La latence du décodeur (i.e. trente-cinq (35) coups d'horloge),
- ✓ La latence du filtre RII complexe (i.e. un coup d'horloge).
- ✓ Le délai d'un coup d'horloge à la sortie du bloc de conjugué.

Le retard des symboles  $x$  doit donc être le double de la latence des symboles  $z$  et  $u$  soit quatre-vingt-six (86) coups d'horloge à deux fois le taux de symbole.

La quantification linéaire des symboles  $x$  permet de réduire le nombre de bits à trois (3) bits par dimension (quantification :  $-2^2+2^1+2^0$ ). Ce bloc sert aussi à calculer le conjugué des symboles  $x$ . La quantification linéaire et le conjugué sont réalisés en même temps avec deux (2) mémoires mortes (une par dimension). Le tableau 3-6 présente la quantification linéaire d'un symbole incluant le conjugué en fonction de sa partie réelle et de sa partie imaginaire.

Tableau 3-6

La quantification linéaire d'un symbole incluant le conjugué.

<b>Entrée partie réelle</b>	<b>Sortie partie réelle</b>	<b>Entrée partie imaginaire</b>	<b>Sortie partie imaginaire</b>
$-\infty, -2.5[$	-3	$-\infty, -2.5[$	3
$[-2.5, -1.25[$	-2	$[-2.5, -1.25[$	2
$[-1.25, 0[$	-1	$[-1.25, 0[$	1
$[0, 1.25]$	1	$[0, 1.25]$	-1
$]1.25, 2.5]$	2	$]1.25, 2.5]$	-2
$]2.5, \infty$	3	$]2.5, \infty$	-3

Dans le même ordre d'idée, la quantification non linéaire des symboles  $z$  permet de réduire le nombre de bits à trois (3) bits par dimension (quantification :  $-2^2+2^1+2^0$ ). La quantification est non linéaire afin de tenir compte de l'effet de l'égaliseur sur les symboles. En effet, ce dernier cherche à regrouper les symboles autour des quatre (4) points de la constellation (i.e.  $\pm 1$  sur chaque dimension). On veut alors une valeur de quantification plus forte si le symbole est loin de cette constellation. La quantification non linéaire est aussi réalisée avec deux (2) mémoires mortes (une par dimension). Le

tableau 3-7 présente la quantification non linéaire d'un symbole en fonction de sa partie réelle et de sa partie imaginaire.

Tableau 3-7  
Quantification non linéaire d'un symbole.

Entrée partie réelle	Sortie partie réelle	Entrée partie imaginaire	Sortie partie imaginaire
$-\infty, -3.25[$	-3	$-\infty, -3.25[$	-3
$[-3.25, -1[$	-2	$[-3.25, -1[$	-2
$[-1, 0[$	-1	$[-1, 0[$	-1
$[0, 1]$	1	$[0, 1]$	1
$]1, 3.25]$	2	$]1, 3.25]$	2
$]3.25, \infty$	3	$]3.25, \infty$	3

Le calcul de l'erreur est réalisé avec deux (2) mémoires vives 256x3 bits (une par dimension). Les mémoires vives sont programmées avec les résultats du calcul de l'erreur quantifiés sur trois (3) bits (quantification :  $-2^2+2^1+2^0$ ) en fonction de la valeur des symboles quantifiés  $z$  et  $u$ . Les avantages de l'utilisation des mémoires vives sont :

- ✓ Une réduction du nombre de cellules logiques programmables requises pour le calcul de l'erreur,
- ✓ La possibilité d'implémenter plusieurs algorithmes d'adaptation sans changer le design du bloc d'adaptation.

Par exemple, l'erreur complexe est calculée au préalable avec l'équation 3-30 si le critère d'adaptation est le MSE, avec l'équation 3-31 si le critère est le CMA ou avec l'équation 3-32 si le critère est le MCMA, les résultats des calculs sont quantifiés sur trois (3) bits par dimension et les mémoires sont programmées avec les résultats quantifiés.

$$e_k = \hat{u}_k - z_k, \quad (3-30)$$

$$e_k = z_k (2 - |z_k|^2), \quad (3-31)$$

$$e_k = z_{k,R} (1 - |z_{k,R}|^2) + j \cdot z_{k,I} (1 - |z_{k,I}|^2). \quad (3-32)$$

Les coefficients du filtre avant et du filtre arrière sont calculés respectivement avec les équations suivantes :

$$c_{k+1} = c_k + \Delta e_k x_k^*, \quad (3-33)$$

$$c_{k+1} = c_k + \Delta e_k u_k^*. \quad (3-34)$$

où  $c$  est le coefficient complexe,  $\Delta$  est le pas d'égalisation,  $x^*$  est le conjugué du symbole dans le filtre avant lié au coefficient  $c$  (i.e.  $c_{-3}, c_{-2}, c_{-1}, c_0$ ),  $u^*$  est le conjugué du symbole dans le filtre arrière lié au coefficient  $c$  (i.e.  $c_1, c_2, c_3$ ) et  $e_k$  est l'erreur calculée à l'aide des mémoires vives. La figure 3-6 présente comment est réalisé le calcul d'un coefficient complexe.

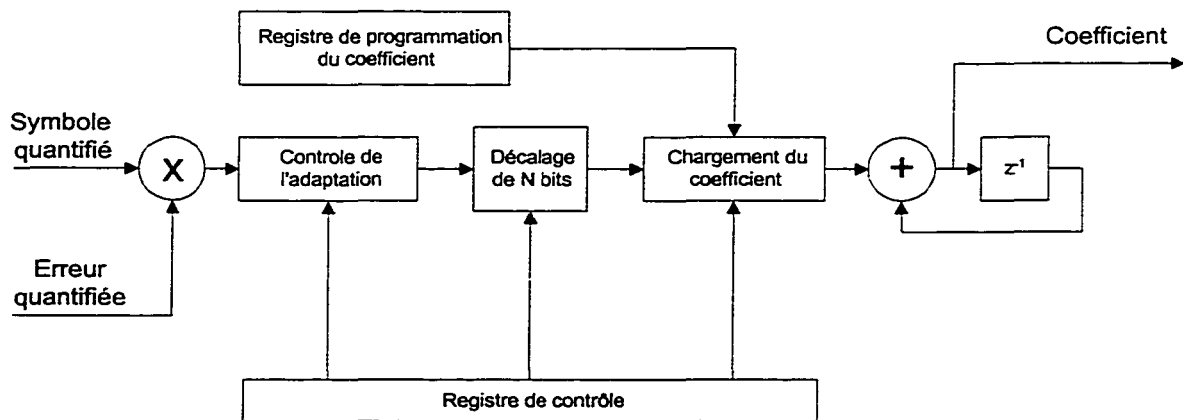


Figure 3-6 Schéma bloc du calcul d'un coefficient complexe



Chaque coefficient complexe est calculé de la même façon. Un registre de contrôle commun à tous les coefficients permet d'utiliser l'égaliseur à retour d'état dans deux (2) modes d'opération. Le premier mode est le mode d'adaptation. Ce mode autorise la mise à jour des coefficients complexes avec un pas d'égalisation programmable. Les valeurs possibles des pas d'égalisation sont données un peu plus loin dans cette section. Le second mode est le mode de non-adaptation. Ce mode empêche la mise à jour des coefficients et la valeur de chacun des sept (7) coefficients peut être ajustée avec un registre de programmation.

Le symbole quantifié est multiplié avec l'erreur quantifiée avec un multiplicateur complexe 3x3 bits pour les coefficients du filtre avant et avec un multiplicateur complexe de 3x1 bits pour les coefficients du filtre arrière. Le résultat de la multiplication est passé à l'étage suivant du calcul si l'égaliseur est en mode d'adaptation. Sinon le résultat est forcé à zéro.

La multiplication par le pas d'égalisation  $\Delta$  est réalisée à l'aide d'un décalage en bloc vers la droite de N bits avec extension de signe (voir le schéma dans l'annexe A). En effet, le décalage vers la droite de N bits est équivalent à une multiplication par  $2^{-N}$ . Cette particularité est intéressante pour éviter la réalisation de multiplicateurs qui sont généralement coûteux en nombre de cellules logiques programmables. Cette étape demande, par contre, vingt-quatre (24) bits par dimension pour conserver la précision du plus petit pas. Le tableau 3-8 présente les quatre (4) pas d'égalisation possibles et le nombre N de bits de décalage nécessaires.

**Tableau 3-8**  
**Pas d'égalisation en fonction du nombre de bits.**

$\Delta$	N
$0.98 \times 10^{-3}$	10 bits
$1.22 \times 10^{-4}$	13 bits
$0.76 \times 10^{-5}$	17 bits
$0.95 \times 10^{-6}$	20 bits

Le résultat de la multiplication par  $\Delta$  est ensuite additionné à la valeur du coefficient à l'instant précédent. L'accumulation se fait sur vingt-quatre (24) bits. Seuls les huit (8) bits les plus significatifs sont présentés à la sortie du bloc d'adaptation. Si le chargement des coefficients est demandé par le registre de contrôle, alors la mémoire de l'accumulateur est remise à zéro et le coefficient dans le registre de programmation est chargé vers l'accumulateur.

De la logique supplémentaire limitant la valeur du coefficient central ( $c_0$ ) est ajoutée. Cette logique permet de protéger l'égaliseur lors d'une mauvaise convergence des coefficients. La partie réelle du coefficient est donc limitée à des valeurs entre 0.5 et 1.5 et la partie imaginaire est limitée à des valeurs entre -0.75 et 0.75.

En terminant, la latence du bloc d'adaptation est quarante-trois (43) coups d'horloge au taux de symbole. La réalisation de ce bloc requiert 900 cellules logiques programmables.

### 3.11 Conclusion

Ce chapitre a présenté comment sont réalisés tous les blocs qui composent l'égaliseur à retour d'état avec décodeur intégré. Chaque composant est initialement conçu avec des portes logiques et/ou de la mémoire. Tous les schémas de réalisation avec le logiciel

Powerview sont fournis à l'annexe A. Le compilateur de Xilinx se sert des schémas de conception logique pour configurer les cellules logiques programmables du circuit FPGA.

Nous avons vu que le nombre de cellules logiques programmables disponibles dans le cadre de ce mémoire est limité à 3136 cellules. La structure à réaliser demande à elle seule environ 2350 CLB. Si on ajoute les 600 CLB nécessaires aux autres composants du récepteur, il reste 186 CLB dans le circuit programmable du récepteur. Ces cellules supplémentaires sont presque toutes requises pour le déverminage du circuit.

Enfin, grâce à la technique du pipeline utilisée dans le filtre avant et dans le bloc d'adaptation, les temps de propagation sont respectés et le récepteur peut fonctionner avec un taux de symbole de 4.096 Msymboles par seconde. Le prochain chapitre présentera les performances, en pratique, du récepteur avec l'égaliseur à retour d'état avec décodeur intégré.

## CHAPITRE 4

### MESURES DES PERFORMANCES

#### 4.1 Introduction

Ce chapitre présente les mesures des performances de l'égaliseur à retour d'état avec décodeur intégré que nous appellerons ici EDI pour simplifier l'écriture. Comme il a été dit dans le chapitre précédent, la radio programmable conçue au LACIME est utilisée pour réaliser l'émetteur et le récepteur (qui inclut l'EDI à tester). Elle est ainsi programmée pour transmettre et recevoir des bits pseudo-aléatoires codés  $R=1/2$  et modulés en QPSK. Cette radio est placée dans l'environnement de test présenté à la figure 4-1.

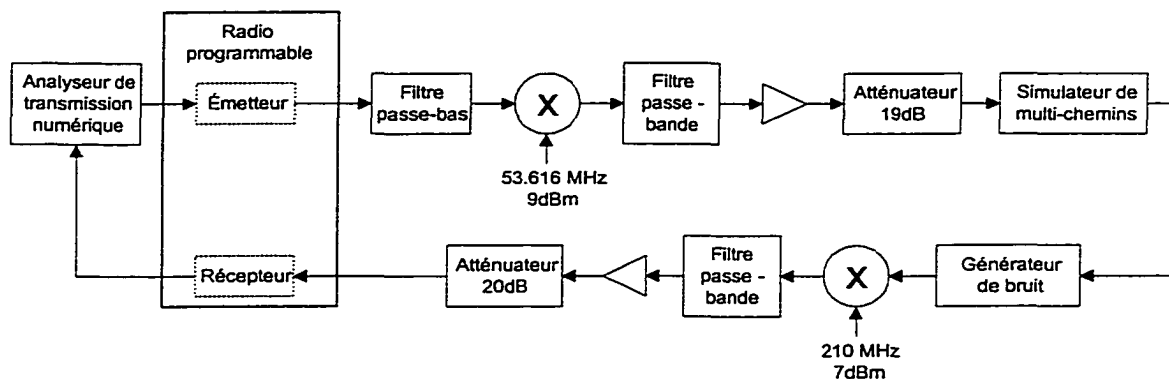


Figure 4-1 Schéma bloc de l'environnement de test.

La séquence pseudo-aléatoire PRBS-23 est générée par l'analyseur de transmission numérique «Digital Transmission Analyzer 3784A» de la compagnie Hewlett Packard.

Cet appareil sert aussi de compteur d'erreurs. Tous les taux d'erreur par bit donnés dans ce chapitre sont calculés par cet appareil.

Le signal modulé par l'émetteur a une fréquence porteuse de 16.384MHz. Ce signal est filtré passe-bas et mélangé avec une porteuse de 53.616MHz pour déplacer le spectre à 70MHz. Il entre ensuite dans un simulateur de propagation multi-chemins («Multipath Fading Simulator 11757B») de la compagnie Hewlett Packard. Ce simulateur permet de créer des évanouissements avec des profondeurs de creux de 0dB à 40dB.

L'appareil suivant permet de compléter le canal tel que présenté dans la section 1.4. Il s'agit d'un générateur de bruit blanc gaussien additif («Noise and Interference Test Set 3708A») aussi de Hewlett Packard. Cet appareil génère le bruit nécessaire selon la fréquence des symboles et selon le rapport  $E_b/N_0$  spécifié.

La dernière étape consiste à mélanger le signal avec une fréquence de 210MHz pour entrer dans la section du récepteur de la radio programmable. Cette section s'attend à recevoir un signal centré à 140MHz pour pouvoir ensuite le centrer autour de 10.976MHz.

Cet environnement de test permet donc d'évaluer les performances de l'égaliseur à retour d'état avec décodeur intégré sous plusieurs conditions :

- ✓ Sans évanouissement (profondeur de creux de 0dB) et sans bruit,
- ✓ Sans évanouissement et avec bruit,
- ✓ Avec évanouissement et sans bruit,
- ✓ Avec évanouissement et avec bruit.

Les performances de l'EDI mesurées dans le cadre de ce mémoire seront données et commentées dans la section suivante.

## 4.2 Mesures des performances

Dans un premier temps, il est intéressant d'évaluer seulement les performances du décodeur lorsque l'égaliseur à retour d'état avec décodeur intégré est en mode de non-adaptation ou en mode d'adaptation. Il est important de noter à ce moment-ci que :

- ✓ En *mode de non-adaptation*, le coefficient central ( $c_0$ ) est fixé à la valeur  $1+0i$  et les autres coefficients sont fixés à la valeur  $0+0i$ ,
- ✓ En *mode d'adaptation*, tous les coefficients sont mis à jour au taux de symbole selon l'algorithme MCMA (pas d'égalisation  $\Delta=1 \times 10^{-6}$ ),
- ✓ Ces paramètres seront maintenus tout au long des tests.

Ainsi, pour évaluer le décodeur uniquement, la profondeur du creux du simulateur de multi-chemins est réglée à 0dB. Le bruit dans le canal est varié en entrant différents ratios  $E_b/N_0$  dans le générateur de bruit. La figure 4-2 de la page suivante donne le taux d'erreur par bit selon le rapport  $E_b/N_0$  pour :

- ✓ L'EDI en mode de non-adaptation,
- ✓ L'EDI en mode d'adaptation,
- ✓ Le récepteur sans l'EDI (QPSK seulement).

De plus, la courbe du taux d'erreur du décodeur à seuil à quantification douce simulé avec le logiciel SPW (sans évanouissement et sans bruit) et la courbe du QPSK théorique sont ajoutées (même courbes que la figure 2-6) pour faciliter la comparaison.

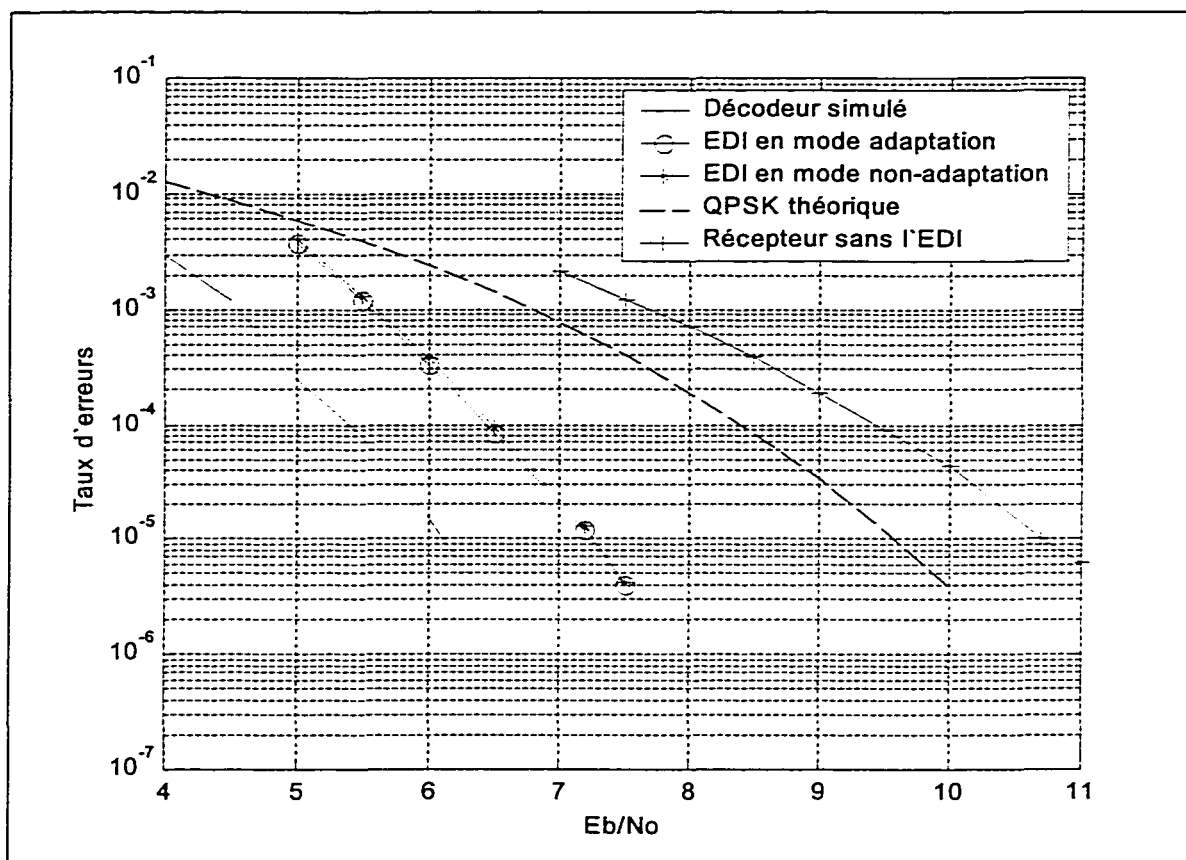


Figure 4-2 Taux d'erreur en fonction de  $E_b/N_0$  sans évanouissement.

A partir de ces courbes, il est possible de déterminer la marge de réalisation de la radio programmable avant même l'égaliseur. Cette marge de réalisation est trouvée en comparant le rapport  $E_b/N_0$  du QPSK théorique avec celui du récepteur sans l'EDI pour un taux d'erreur de  $1 \times 10^{-5}$ . La différence est d'environ 1.1dB. Cette différence était déjà connue avant la réalisation de ce mémoire.

On retrouve cette différence de 1.1dB entre la courbe du taux d'erreur du décodeur à seuil simulé et la courbe du taux d'erreur de l'EDI en mode de non-adaptation. Ce résultat permet de croire que le décodeur fonctionne parfaitement puisque cette différence est expliquée par la marge d'implémentation de la radio programmable.

Enfin, on peut noter une légère différence entre la courbe du taux d'erreur de l'EDI en mode d'adaptation et la courbe du taux d'erreur de l'EDI en mode de non-adaptation. Ce résultat est intéressant puisqu'il montre que l'égaliseur corrige un filtrage non optimal entre l'émetteur et le récepteur et donne ainsi un gain de 0.1dB.

En second lieu, il est pertinent d'évaluer l'influence de l'égaliseur à retour d'état avec décodeur intégré en mode de non-adaptation et en mode d'adaptation sur un signal sans bruit mais détérioré par un évanouissement au centre de son spectre. Le meilleur moyen pour démontrer l'effet de l'égalisation sur un signal est l'observation des constellations avant et après l'égaliseur. Le simulateur de multi-chemins est réglé à une profondeur de creux de 34dB au centre du spectre (fréquence 70MHz) et le générateur de bruit blanc est mis hors fonction. Les figures 4-3 et 4-4 présentent respectivement la constellation à l'entrée de l'égaliseur et la constellation à la sortie de la somme des symboles du filtre avant et du filtre arrière (voir la figure 3-1).

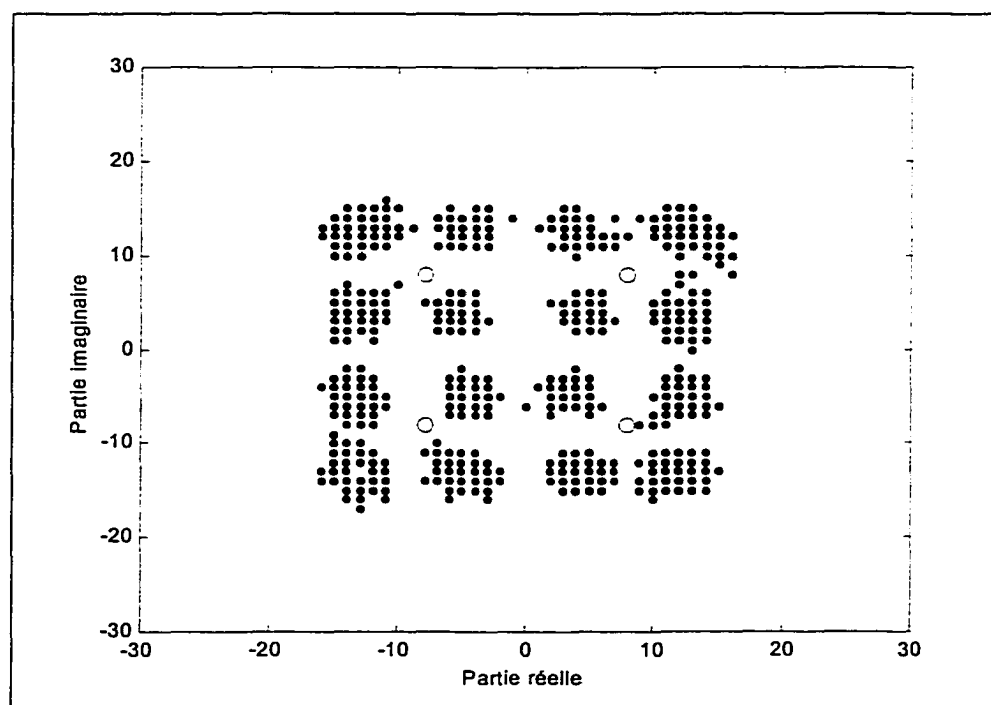


Figure 4-3 Constellation QPSK avant l'égaliseur (creux 34dB à 70MHz, sans bruit).



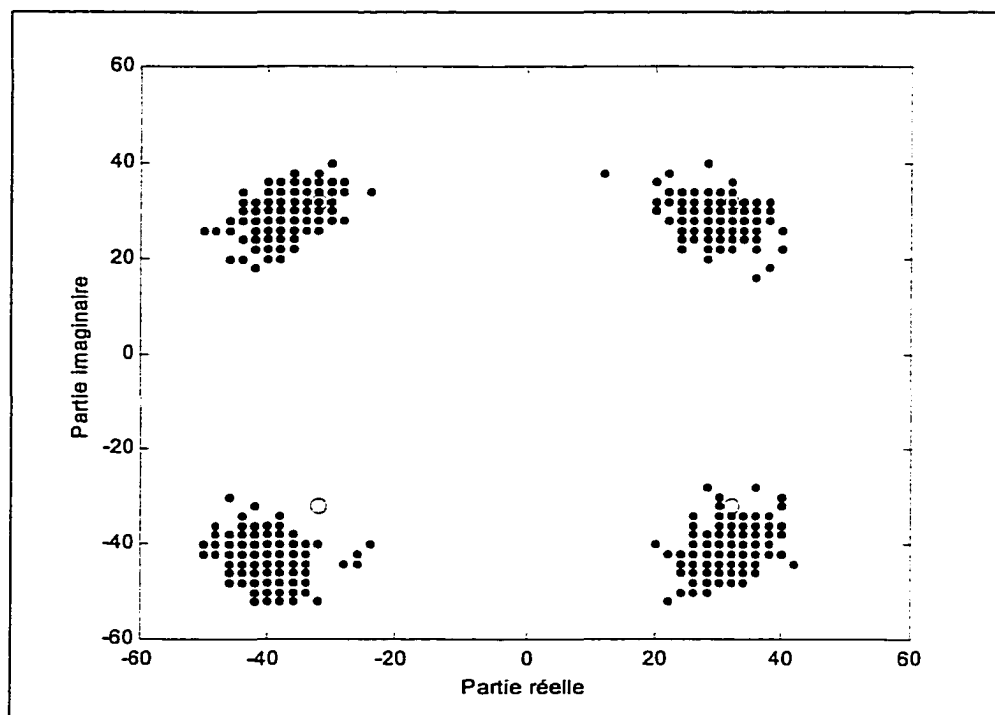


Figure 4-4 Constellation QPSK après l'égaliseur (creux 34dB à 70MHz, sans bruit).

La figure 4-3 illustre bien les effets de l'interférence intersymbole sur le signal à l'entrée de l'égaliseur. En effet, les "o" représentent la constellation idéale (celle sans bruit et sans interférence intersymbole). Or, il y a quatre (4) nuages de points autour de chacun des points de la constellation idéale. Ces nuages sont le résultat de la propagation multi-chemins.

La constellation à la sortie de la somme des symboles du filtre avant et du filtre arrière de l'EDI en mode de non-adaptation n'est pas montrée ici car elle est exactement identique à celle présentée à la figure 4-3. Ceci est normal puisque le coefficient central (fixé à la valeur  $1+0i$ ) laisse passer le signal sans le modifier et que le filtre arrière est absent (i.e. ces coefficients ont la valeur  $0+0i$ ). Le taux d'erreur à la sortie du récepteur est zéro. Ceci s'explique par le fait que même avec les interférences intersymboles, les nuages de points sont situés à l'intérieur des limites des régions de décision (les axes).

D'autre part, la constellation à la sortie de la somme des symboles du filtre avant et du filtre arrière de l'EDI en mode d'adaptation (figure 4-4) est intéressante parce qu'on voit bien l'effet de l'égalisation sur le signal. En effet, on retrouve un seul nuage de points autour de chacun des points de la constellation idéale. L'algorithme MCMA force donc la sortie de l'égaliseur (ou l'entrée du décodeur) à former une constellation qui ressemble à la constellation transmise à l'émetteur. Il est clair dans ce cas que le taux d'erreur à la sortie du décodeur est zéro.

Par contre, la figure 4-4 montre un décalage important des nuages de points situés dans les quadrants II, III et IV (le quadrant I étant celui formé des axes positifs). Le centre de ces nuages devrait être le point de la constellation idéal représenté par un "o". Une hypothèse qui reste à vérifier pour expliquer ce décalage est le traitement du zéro lors des quantifications linéaires et non linéaires des symboles. En effet, à cause du choix du format complément à deux, le zéro est toujours considéré comme un nombre positif lors de la quantification. Ce biais pourrait être la cause du décalage dans trois (3) quadrants sur quatre (4).

Dans un troisième temps, les performances de l'égaliseur à retour d'état avec décodeur intégré sont évaluées pour différentes fréquences de l'évanouissement et sans bruit. Pour ce faire, la fréquence de l'évanouissement dans le simulateur de multi-chemins est réglée avec des valeurs de 68.8MHz à 71.4MHz. Les courbes W sont ensuite tracées pour les trois cas qui nous intéressent (i.e. l'EDI en mode de non-adaptation, l'EDI en mode d'adaptation et le récepteur sans l'EDI). La figure 4-5 présente les courbes W.

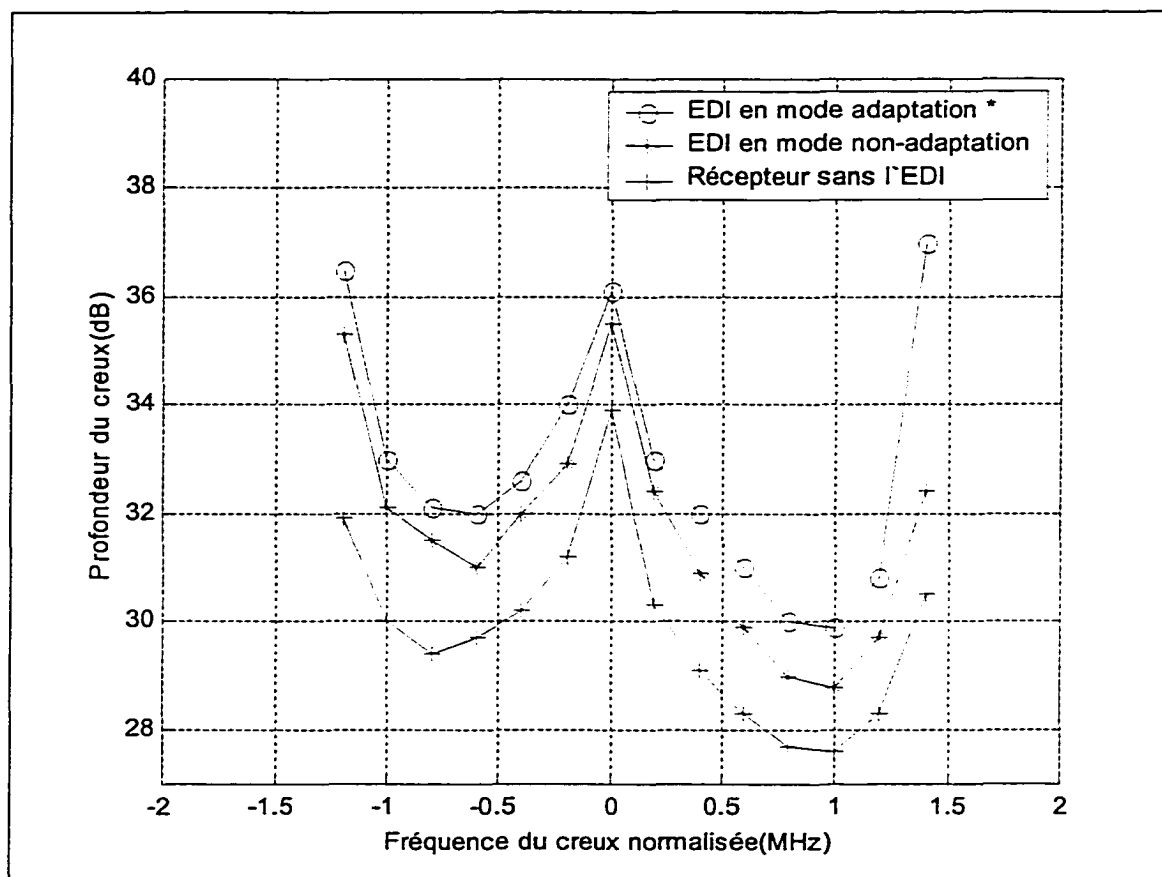


Figure 4-5 Courbes W sans bruit.

Il est important de noter que les régions à l'intérieur des W correspondent aux régions pour lesquelles le taux d'erreur dépasse le seuil de  $1 \times 10^{-5}$  et ce pour la courbe de l'EDI en mode de non-adaptation et pour la courbe du récepteur sans l'EDI. Par contre, la région à l'intérieur du W pour l'EDI en mode d'adaptation représente un taux d'erreur nulle et sans perte de synchronisation (i.e. au niveau de l'analyseur de transmission numérique).

En effet, il a été constaté, lors de la prise des résultats, que de la gigue au niveau de la porteuse et au niveau de l'horloge causait des rotations soudaines de la constellation et,

par le fait même, engendrait des pertes de synchronisation de l'analyseur de transmission numérique. Cette gigue est un des effets indésirables de l'interférence intersymbole [22].

Normalement, les algorithmes de recouvrement de porteuse et d'horloge utilisent les symboles égalisés pour prévenir ce genre de problème [11][21][22][23]. Comme il est montré dans la figure 3-1, les blocs de recouvrement de porteuse et d'horloge de ce mémoire utilisent les symboles avant l'égaliseur. Ce choix était fixé dès le début du projet, et ce pour permettre la réalisation et le déverminage de l'égaliseur à retour d'état avec décodeur intégré. Évidemment, les performances globales du récepteur sont limitées dans ces conditions.

Malgré ce fait, nous pouvons constater que l'EDI en mode d'adaptation ne donne tout de même aucune erreur à des profondeurs de creux supérieures à l'EDI en mode de non-adaptation et au récepteur sans l'EDI (qui eux donnent un taux d'erreur de  $1 \times 10^{-5}$ ). La courbe W de l'EDI en mode d'adaptation est d'au moins 0.5dB supérieure à la courbe W de l'EDI en mode de non-adaptation, qui elle-même est d'au moins 1.5dB supérieure au récepteur sans l'EDI.

Dans un autre ordre d'idées, nous allons maintenant mesurer les performances de l'égaliseur à retour d'état avec décodeur intégré dans des conditions d'évanouissement et de bruit. La figure 4-6 présente le taux d'erreur en fonction du rapport  $E_b/N_0$  pour l'EDI en mode de non-adaptation, l'EDI en mode d'adaptation et le récepteur sans l'EDI en présence d'un évanouissement de 28dB au centre du spectre (fréquence 70MHz). La courbe de performances en simulation du décodeur à seuil (sans évanouissement et sans bruit) est aussi ajoutée sur la figure pour aider à la comparaison.

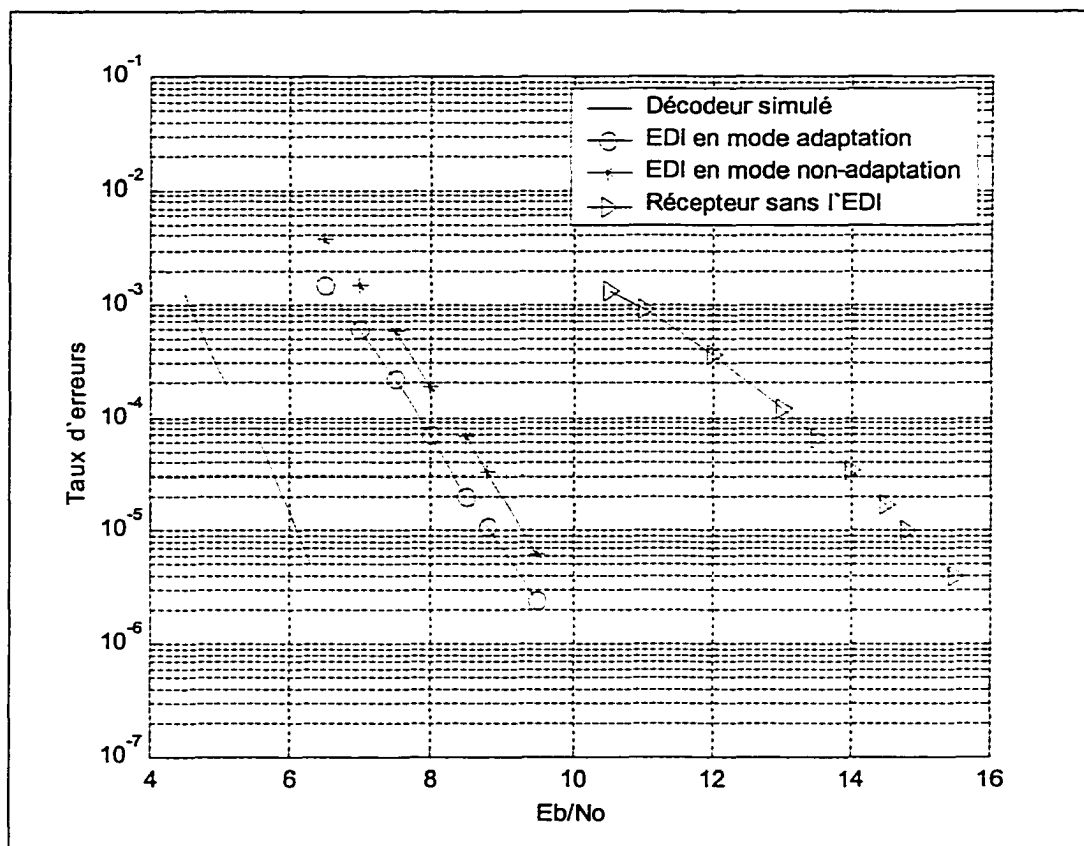


Figure 4-6 Taux d'erreur en fonction de  $E_b/N_0$  (creux 28dB à 70MHz).

Pour un taux d'erreur de  $1 \times 10^{-5}$  et en comparant avec le décodeur théorique, nous notons une diminution des performances à la sortie du décodeur de 3dB lorsque l'EDI fonctionne en mode de non-adaptation. Cependant, un gain de 5dB est quand même réalisé sur le récepteur sans l'EDI. Ces deux courbes illustrent donc l'effet du décodeur seulement. Il est intéressant de constater que ce dernier interprète l'interférence intersymbole comme du bruit et le corrige dans une certaine mesure. D'un autre côté, lorsque l'EDI est en mode d'adaptation, un gain supplémentaire de 0.5dB est réalisé. Ce gain est alors un effet direct et unique de l'égalisation.

Pour mieux illustrer l'influence de l'EDI sur la constellation du signal, la profondeur du creux de l'évanouissement est augmentée à 34dB (fréquence 70Mhz) et on recherche

un taux d'erreur de  $1 \times 10^{-5}$  en variant le bruit lorsque l'EDI est en mode de non-adaptation. Le rapport  $E_b/N_0$  trouvé est alors de 16.6dB. La figure 4-7 présente la constellation à l'entrée de l'EDI pour un évanouissement de 34dB et pour un ratio  $E_b/N_0$  de 16.6dB.

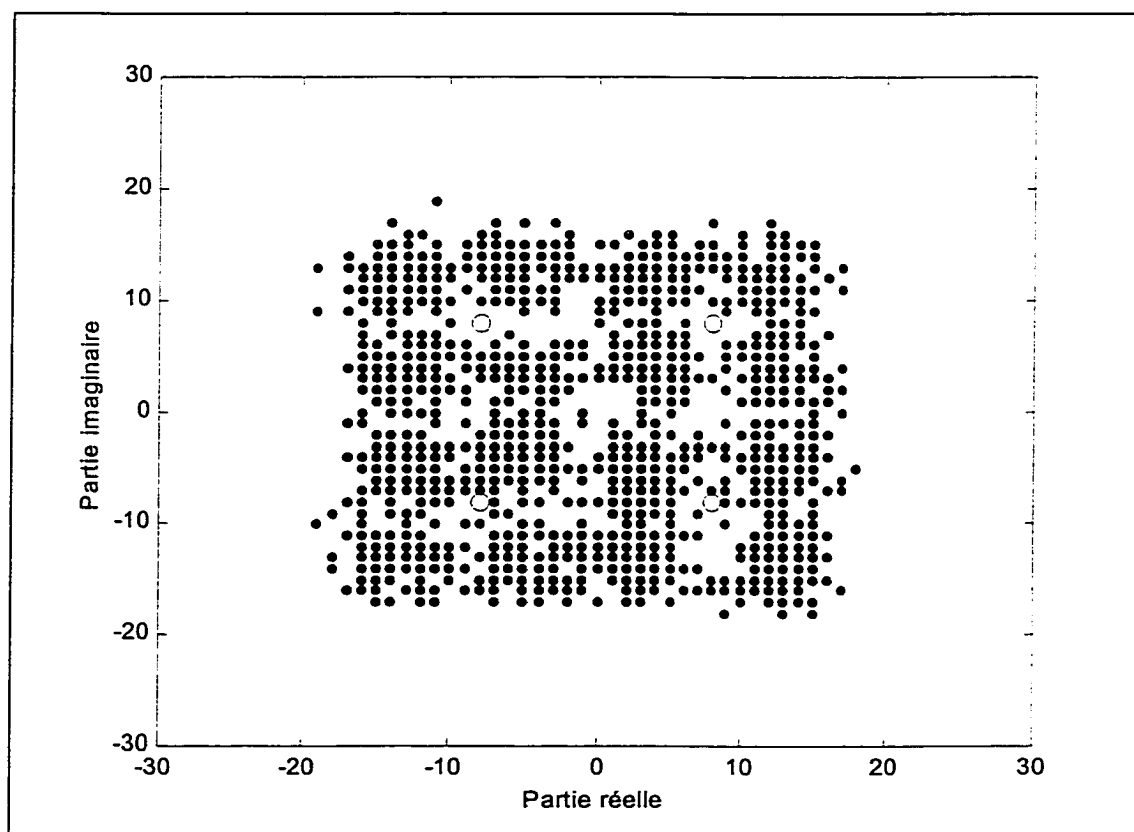


Figure 4-7 Constellation QSPK avant l'égaliseur (creux 34dB à 70MHz,  $E_b/N_0$  16.6dB).

Cette constellation se retrouve également à l'entrée du décodeur à seuil puisque nous sommes en mode de non-adaptation et c'est elle qui cause le taux d'erreur de  $1 \times 10^{-5}$ . En effet, on voit bien les points qui sont situés sur les axes (i.e. les régions de décision) et qui sont sujets à de mauvaises décisions de la part du décodeur. Si on règle maintenant l'EDI en mode d'adaptation, la constellation devient celle présentée dans la figure 4-8.

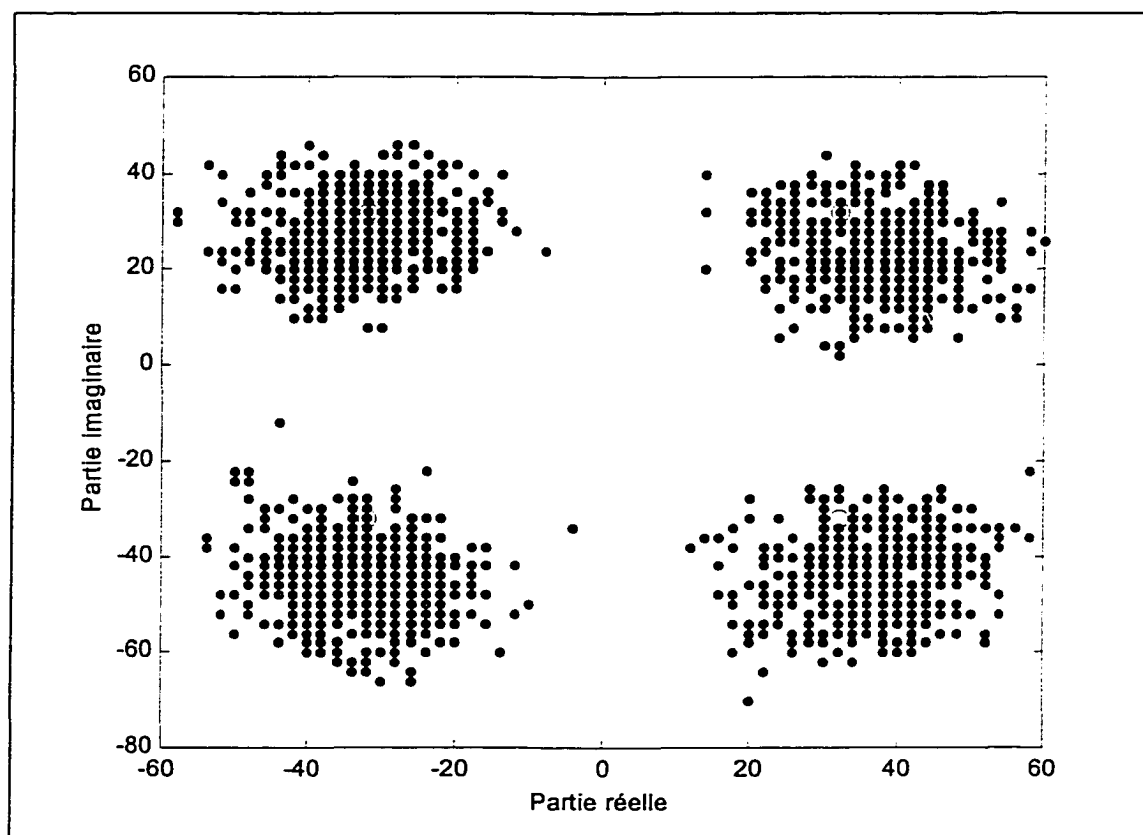


Figure 4-8 Constellation QSPK après l'égaliseur (creux 34dB à 70MHz,  $E_b/N_0$  16.6dB).

Cette constellation donne maintenant un taux d'erreur de  $1 \times 10^{-7}$ . L'adaptation des coefficients a permis de passer d'un taux d'erreur de  $1 \times 10^{-5}$  à un taux d'erreur de  $1 \times 10^{-7}$ . On remarque bien les nuages de points autour de chacun des points de la constellation idéale. Les erreurs ici sont probablement causées par le décalage des nuages déjà remarqué dans la figure 4-4 (avec une rotation de  $90^\circ$ ). On voit en effet des points qui sont proches des axes dans le premier quadrant. Le centre de ce nuage est trop bas si on le compare avec le centre idéal.

A présent, il est intéressant de trouver à quel rapport  $E_b/N_0$ , l'EDI en mode d'adaptation donne un taux d'erreur semblable de  $1 \times 10^{-5}$ . Malheureusement et encore une fois, des pertes de synchronisation (causées par la gigue de la porteuse et de

l'horloge) nous empêchent de trouver ce rapport. Le dernier rapport  $E_b/N_0$  qui permet à l'analyseur de trouver un taux d'erreur sans perte de synchronisation est 14.7dB. A ce rapport de signal à bruit, le taux d'erreur est encore plus petit que  $1 \times 10^{-5}$ . La figure 4-9 donne la constellation à l'entrée de l'égaliseur dans ce cas.

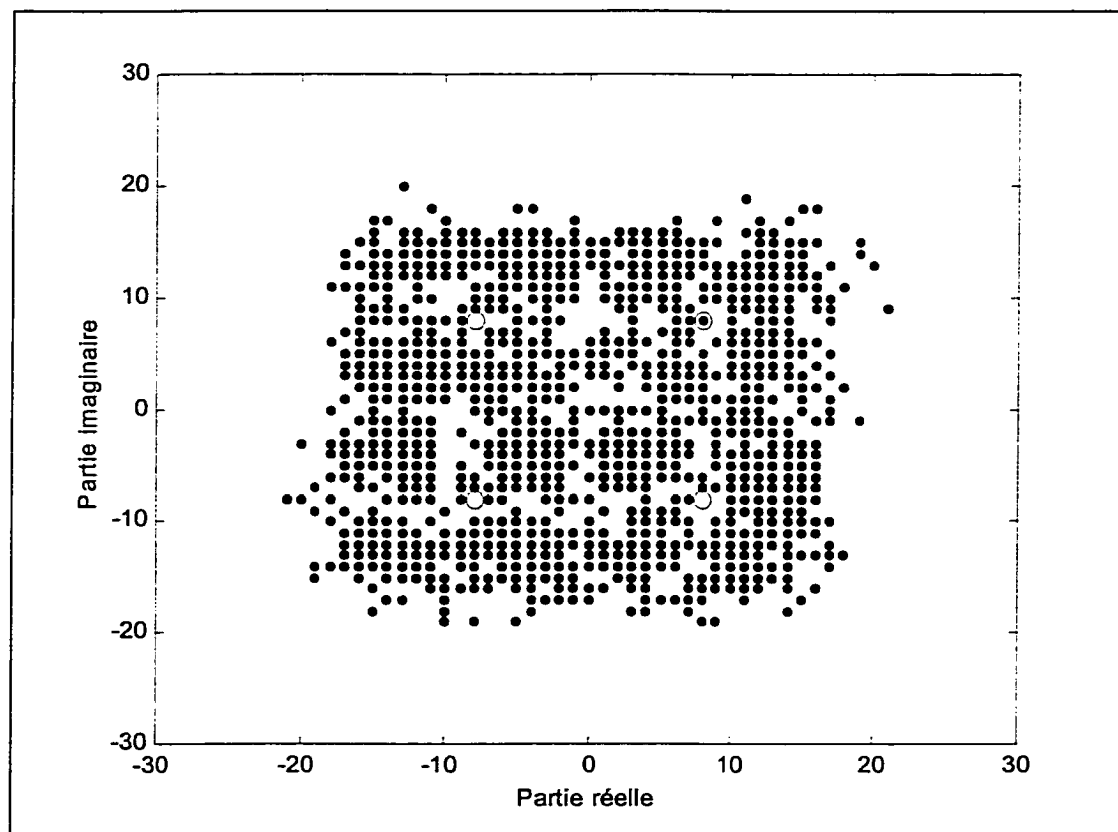


Figure 4-9 Constellation QSPK avant l'égaliseur (creux 34dB à 70MHz,  $E_b/N_0$  14.7dB).

Il est évident que cette constellation contient plus de bruit que la constellation de la figure 4-7. Le taux d'erreur en mode de non-adaptation est d'ailleurs plus élevé. Il se situe à  $1 \times 10^{-4}$ . La figure 4-10 présente la constellation à la sortie de la somme des symboles du filtre avant et du filtre arrière lorsque l'EDI est en mode d'adaptation.



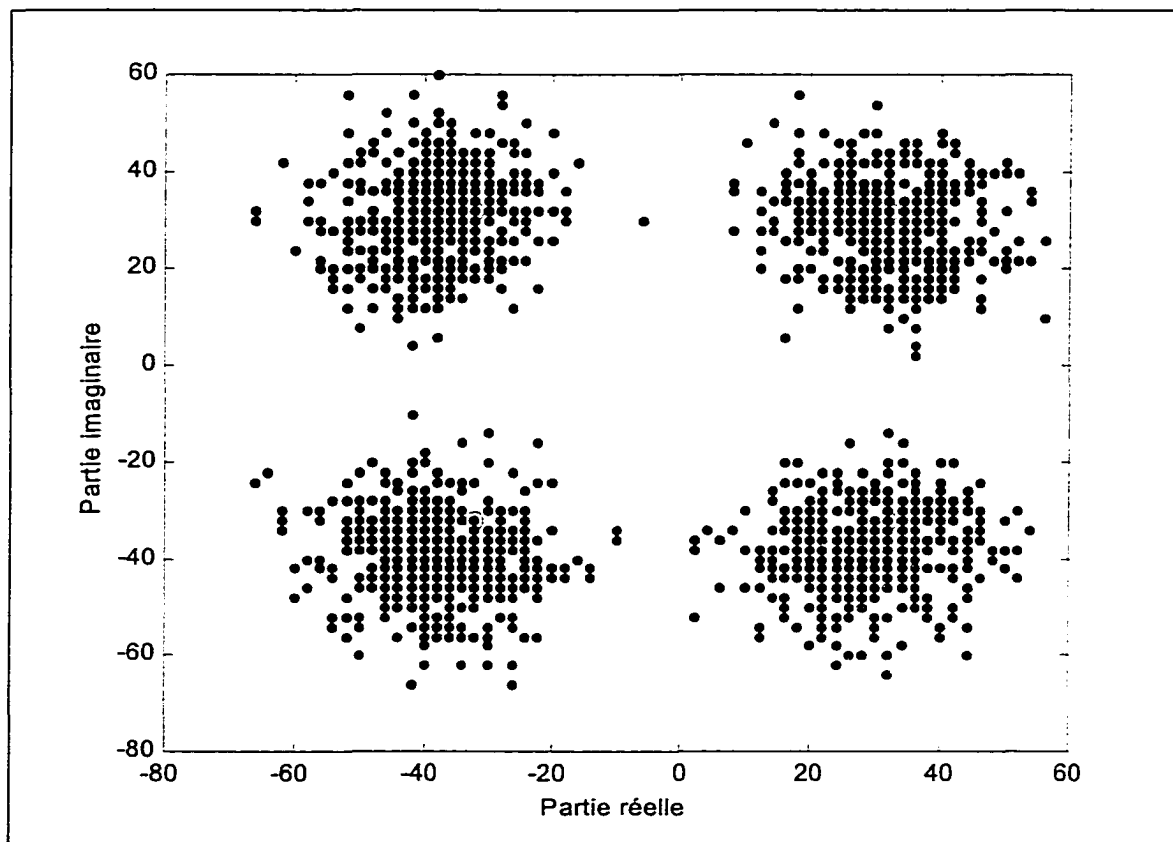


Figure 4-10 Constellation QSPK après l'égaliseur (creux 34dB à 70MHz,  $E_b/N_0$  14.7dB).

Malgré le niveau de bruit plus élevé, l'EDI regroupe encore les points en nuage autour de chaque point de la constellation idéale. Avec cette constellation, on comprend bien pourquoi le taux d'erreur de l'analyseur est plus petit que  $1 \times 10^{-5}$ . En fait, il devrait être semblable au taux d'erreur de la figure 4-8, c'est-à-dire  $1 \times 10^{-7}$ , mais la gigue de phase de l'horloge et de la porteuse cause des erreurs.

### 4.3 Conclusion

Au cours de ce chapitre, nous avons vu les performances de l'égaliseur à retour d'état à décodeur intégré. Pour déterminer les performances, la radio programmable était placée dans un environnement de test composé, entre autre chose, d'un simulateur de

propagation multi-chemins et d'un générateur de bruit. Cet environnement de test a permis de soumettre le récepteur (et par le fait même l'égaliseur à retour d'état avec décodeur intégré) à différentes caractéristiques de canal. Ainsi, le récepteur fut testé :

- ✓ Seulement en présence de bruit blanc gaussien,
- ✓ Seulement avec un évanouissement dans son spectre,
- ✓ Avec un évanouissement et du bruit.

Les performances exprimées dans ce chapitre permettent d'affirmer que la réalisation de l'égaliseur à retour d'état avec décodeur intégré est possible et fonctionnelle. En effet, les performances de l'EDI en mode d'adaptation sont meilleures que les performances de l'EDI en mode de non-adaptation et elles sont aussi meilleures que les performances du récepteur sans l'EDI. Ceci est vrai pour toutes les conditions auxquelles le récepteur fut soumis.

Par contre, les performances n'ont pu être complétées à cause d'un problème externe à l'égaliseur, c'est-à-dire la gigue de la porteuse et de l'horloge lors d'évanouissements profonds. Ce problème était prévisible à cause des algorithmes de recouvrement de la porteuse et de l'horloge qui ne fonctionnent pas bien dans ces conditions.

D'autre part, les performances sont sûrement quelque peu limitées par le décalage des nuages de points observé en mode d'adaptation. Les hypothèses pour expliquer ce phénomène sont : un mauvais traitement du zéro lors de la quantification et le format complément à deux (qui ne contient pas de «zéro négatif»). Ces hypothèses doivent cependant être vérifiées et ce problème doit être corrigé.

Somme toute, il a été démontré que l'égaliseur à retour d'état avec décodeur intégré permet de prévenir, dans une certaine mesure, les erreurs causées par le bruit et les interférences intersymboles.

## CONCLUSION

Nous avons vu, dans ce mémoire, qu'un système de communication typique est composé de trois principaux composants : l'émetteur, le canal et le récepteur. L'émetteur est constitué du codeur de la source d'information, du codeur de canal, du modulateur et du filtre de mise en forme. Toutes ces opérations ont pour but de protéger le signal d'information contre le bruit et les interférences que le canal engendre inévitablement.

En effet, nous avons vu que le canal est un milieu physique sujet au bruit thermique et aux interférences intersymboles. En radiotélécommunication, plus particulièrement, les interférences intersymboles sont le résultat d'une propagation multi-chemins.

L'information ainsi corrompue doit être estimée par le récepteur. Ce dernier regroupe le filtre de blanchiment, l'égaliseur, le démodulateur, le décodeur de canal et le décodeur de source. Généralement, chacun de ces composants fonctionne d'une façon indépendante l'un de l'autre.

Plusieurs recherches ont été faites sur les deux principaux composants du système de communication numérique soit l'égaliseur et l'ensemble codeur/décodeur. Le but de ces recherches était toujours le même : produire une estimation de l'information reçue la plus exacte et la plus fidèle à l'information transmise. Parmi les plus récentes, on retrouve celle de Ladouceur en 1999. Cette étude proposait de combiner partiellement un décodeur à seuil à même la boucle d'un égaliseur à retour d'état. Selon les résultats de cette recherche, la combinaison des deux composants permet d'obtenir une information plus fiable à l'entrée du filtre arrière de l'égaliseur et ainsi de limiter la propagation d'erreurs pour un rapport signal à bruit de faible intensité.

La recherche de Ladouceur fut donc la référence de base de ce travail. En effet, l'objectif de ce mémoire était, dans un premier temps, de réaliser l'égaliseur à retour d'état avec décodeur intégré. La réalisation matérielle fut possible grâce à une radio programmable développée au Laboratoire de communication et d'intégration de la micro-électronique de l'École de technologie supérieure.

Ce mémoire a présenté comment sont conçus et réalisés tous les blocs qui composent l'égaliseur à retour d'état avec décodeur intégré. Chaque composant a été initialement réalisé avec des portes logiques et/ou de la mémoire. Le compilateur de Xilinx utilisait ensuite les schémas de conception logique pour configurer les cellules logiques programmables du circuit FPGA.

Nous avons vu que le nombre de cellules logiques programmables disponibles dans le cadre de ce mémoire était limité à 3136 cellules à cause du type de circuit programmable utilisé pour le récepteur. L'égaliseur à retour d'état avec décodeur intégré a requis environ 2350 CLB. Si on ajoute les 600 CLB nécessaires aux autres composants du récepteur, il restait 186 CLB pour le déverminage du circuit. Par ailleurs, la technique de pipeline utilisée dans le filtre avant et dans le bloc d'adaptation a permis de faire fonctionner la radio programmable à un taux de symbole de 4.096 Msymboles par seconde.

La seconde partie de l'objectif de ce mémoire était de mesurer les performances du récepteur incluant l'égaliseur à retour d'état avec décodeur intégré dans un environnement de test. Cet environnement de test était composé principalement d'un simulateur de propagation multi-chemins et d'un générateur de bruit. Le récepteur fut ainsi testé dans différentes conditions de bruit et d'évanouissement.

Les performances exprimées dans ce mémoire permettent de conclure que la réalisation de l'égaliseur à retour d'état avec décodeur intégré est possible et fonctionnelle. Par contre, les performances n'ont pu être complétées à cause d'un problème externe à l'égaliseur. En effet, la gigue de l'horloge et de la porteuse lors d'un évanouissement profond empêchait la lecture du taux d'erreur dans cette condition. Des rotations soudaines de la constellation causaient des pertes de synchronisation au niveau de l'analyseur de transmission numérique. Ce problème était prévisible car les symboles utilisés dans les boucles de recouvrement de l'horloge et de la porteuse ne provenaient pas de l'égaliseur et étaient donc susceptibles aux interférences intersymboles. Ce choix était voulu pour permettre le développement du prototype.

D'autre part, les performances du récepteur sont sûrement quelque peu limitées par le décalage des nuages de points observé en mode d'adaptation. Un mauvais traitement du zéro lors de la quantification et le format complément à deux (qui ne contient pas de «zéro négatif») sont des hypothèses pour expliquer ce décalage qui doivent être vérifiées. Une chose est certaine, le problème doit être corrigé.

Plusieurs possibilités n'ont malheureusement pas été exploitées avec ce projet, et ce à cause du temps limité pour la remise du mémoire. Parmi quelques-unes, on retrouve la possibilité de réaliser l'algorithme LMS ou l'algorithme CMA à même la structure réalisée dans ce projet. En effet, les mémoires vives dans le bloc d'adaptation peuvent être programmées pour calculer l'erreur selon le critère du MSE ou le critère du CMA.

D'autres mesures de performances pourraient également être faites sur l'égaliseur réalisé dans ce mémoire, et ce en variant le pas d'égalisation. Ce pas est effectivement programmable, et il peut avoir des valeurs de  $1 \times 10^{-3}$  à  $1 \times 10^{-6}$ . Il serait intéressant de voir, en pratique, l'influence du pas d'égalisation sur la convergence des coefficients.

En terminant, ce mémoire a démontré qu'il est possible de réaliser, dans un circuit programmable, un égaliseur à retour d'état avec un décodeur intégré dans sa boucle de rétroaction. Quelques correctifs restent à être apportés sur le récepteur pour atteindre les performances optimales mais les possibilités offertes par cette réalisation sont nombreuses.

## BIBLIOGRAPHIE

1. Proakis, J. G. (1995). Digital communications (3<sup>ième</sup> éd.). New York : McGraw-Hill.
2. Lin, S., Costello, D. J. (1983). Error control coding. Englewood Cliffs : Prentice-Hall.
3. Smith, D. R. (1993). Digital transmission systems (2<sup>ième</sup> éd.). New York : Chapman & Hall.
4. Elias, P. (1955). Coding for Noisy Channels. IRE Conv rec, Part 4, pp. 37-47.
5. Massey, J. L. (1963). Threshold Decoding. MIT Press, Cambridge.
6. Stremler, F. G. (1990). Introduction to communication systems (3<sup>ième</sup> éd.). Addison-Wesley.
7. Ziemer, R. E., Tranter, W. H. (1995). Principles of communications (4<sup>ième</sup> éd.). New York : John Wiley & Sons.
8. Belfiore, C. A., Park, J. H. (1979). Decision Feedback Equalization. Proceedings of the IEE, volume 67, no 8, pp 1143-1156.
9. Qureshi, S. U. H. (1985). Adaptive Equalization. Proceedings of the IEEE, volume 73, no 9, pp 1349-1384.
10. Ladouceur, S. (1999). Conception d'un égaliseur numérique. École de technologie supérieure, Montréal.
11. Godard, D. N. (1980). Self-Recovering Equalization and Carrier Tracking in Two-Dimensional Data Communication Systems. IEEE transactions on communications, volume 28, no 11, pp 1867-1875.
12. Litwin, L. R., Endres, T. J., Hulyalkar, S. N., Zoltowski, M. D. (1999). Proposed Bit Precisions for a VLSI Implementation of the Constant Modulus Algorithm. 1999 2<sup>nd</sup> IEEE Workshop on Signal Processing Advances in Wireless Communications, pp 300-303.

13. Lavoie, P., Haccoun, D., Savaria, Y. (1991). New VLSI Architectures for Fast Soft-Decision Threshold Decoders. IEEE transactions on communications, volume 39, no 2, pp 200-207.
14. Gagnon, F., Batani, N., Dam, T. Q. (1995). Simplified Designs for AAPP Soft Decision Threshold Decoders. IEEE transactions on communications, volume 43, no 2/3/4, pp 743-750.
15. Wozencraft, J. M., Jacobs, I. M. (1965). Principles of Communication Engineering. New York : John Wiley&Sons.
16. Proakis, J. G., Manolakis, D. G. (1996). Digital Signal Processing (3<sup>ième</sup> éd.). Upper Saddle River : Prentice-Hall.
17. Honig, M. L. Messerschmitt, D. G., (1984). Adaptative Filters : Structures, Algorithms and Applications. Boston : Kluwer Academic Publishers.
18. Gitlin, R. D., Hayes, J. F., Weinstein, S. B. (1992). Data Communications Principles. New York : Plenum Press.
19. Xilinx (1999). XC400E and XC4000X Series Field Programmable Gate Arrays.
20. Letocha, J. (1985). Introduction aux circuits logiques (2<sup>ième</sup> édition). Montréal : McGraw-Hill.
21. Oh, K. N., Chin, Y. O. (1995). Modified Constant Modulus Algorithm : Blind Equalization and Carrier Phase Recovery Algorithm. IEEE International Conference on Communications, volume 1, pp 498-502.
22. Noguchi, T., Daido, Y., Nossek, A. (1986). Modulation Techniques for Microwave Digital Radio. IEEE Communications Magazine, volume 24, no 10, pp 21-30.
23. Treichler, J. R., Larimore, M. G., Harp, J. C. (1998). Practical Blind Demodulators for High-Order QAM Signals. Proceedings of the IEEE, volume 86, no 10, pp 1907-1926.
24. Johnson, C. R., Schniter, P., Endres, T. J., Behm, J.D., Brown, D.R., Casas, R. A. (1998). Blind Equalization Using the Constant Modulus Criterion : A Review. Proceedings of the IEEE, volume 86, no 10, pp 1927-1949.

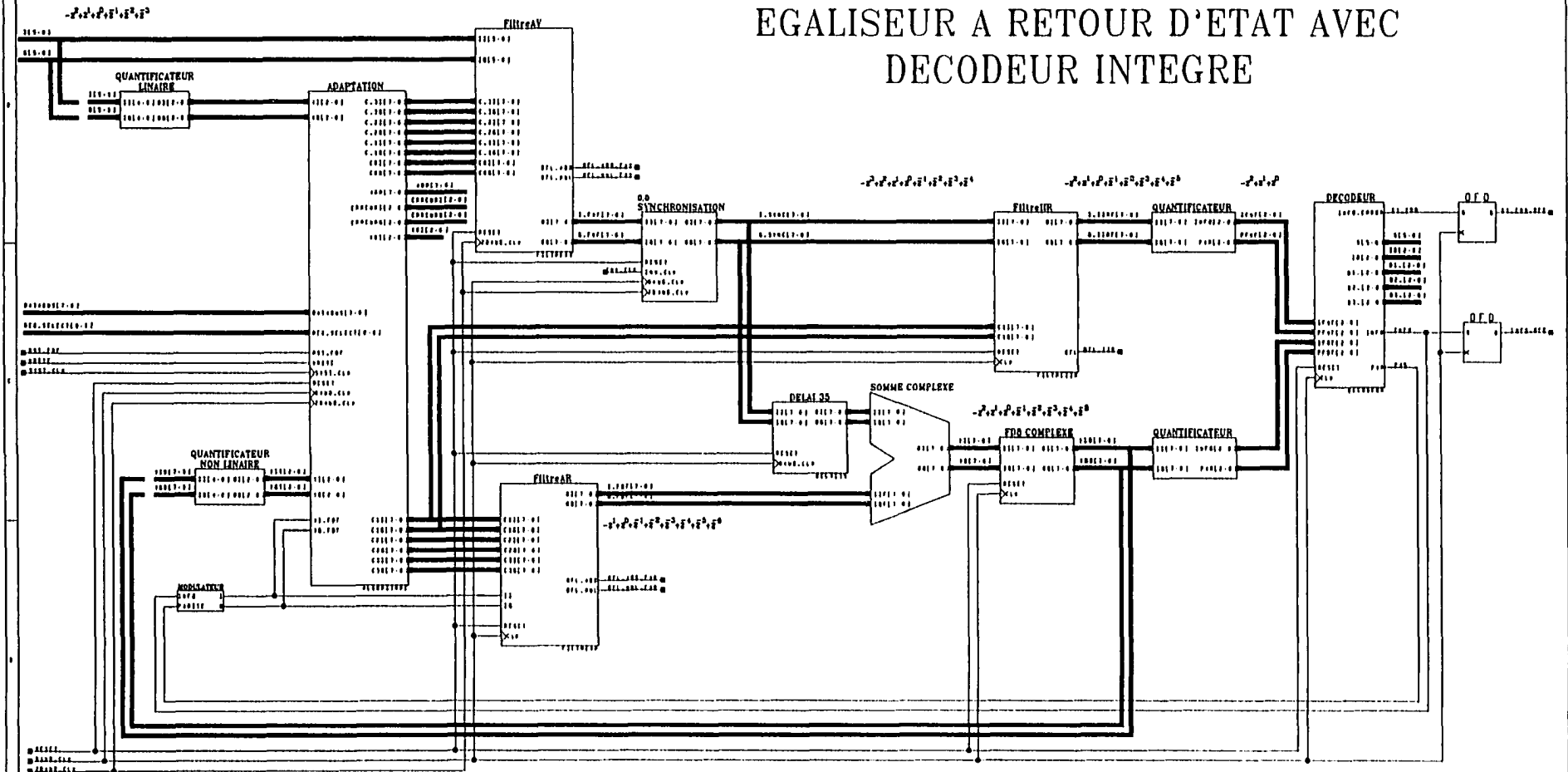


25. Widrow, B., Hoff, M. E., (1960). Adaptative Switching Circuits. IRE WESCON Conv. Rec., pt 4, pp 96-104.

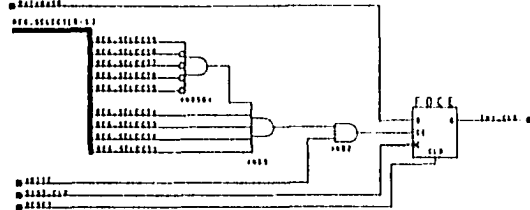
## **ANNEXE A**

### **LES SCHÉMAS LOGIQUES DE RÉALISATION**

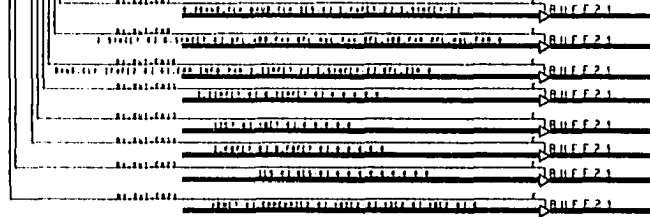
# EGALISEUR A RETOUR D'ETAT AVEC DECODEUR INTEGRE



## CHOIX DU SYMBOLE



X0(0-4), X3(7-0), X2(7-0)



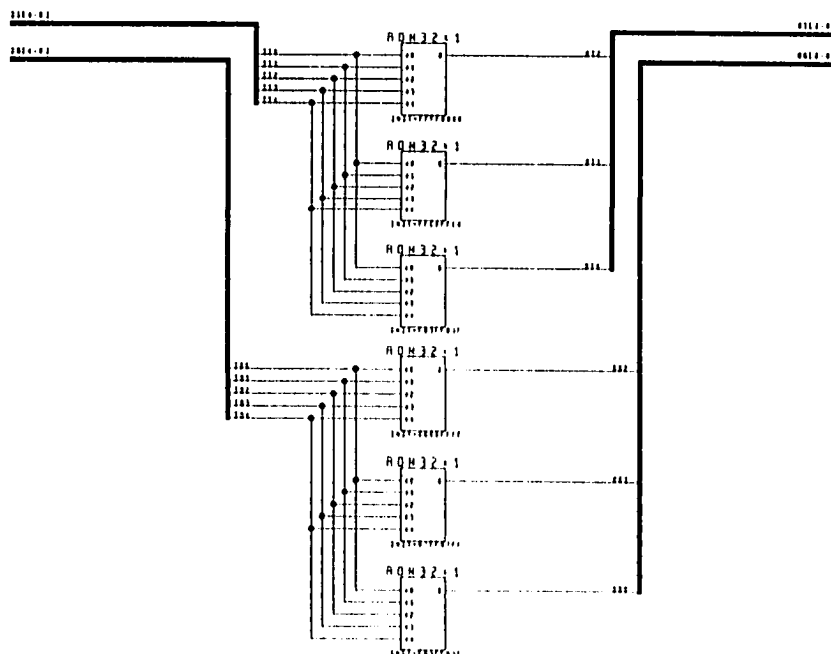
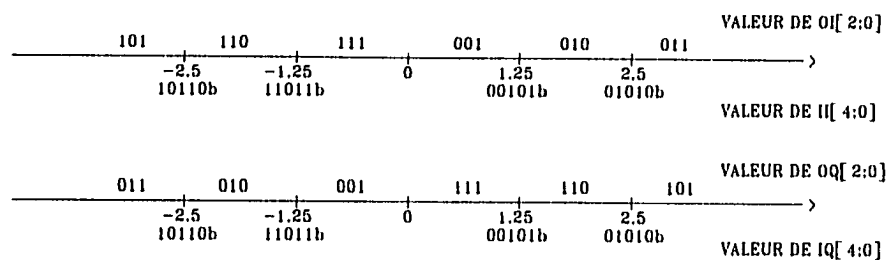
LACIME	
ECOLE DE TECHNOLOGIE SUPERIEURE	
TITRE DU PROJET: HCLIS/SPAS SYSTEM	
DESCRIPTION DE LA PAGE: EGALISEUR COMPLET	
CODE DE PROJET: PFI_0001	NOM: EGALISEUR
CODE D'OBJET: PFI_0001R	DATE: 02/05/2000

# QUANTIFICATION LINEAIRE DE I ET Q AVEC CONJUGUE

QUANTIFICATION DES ENTREES II ET IQ :  $-2^2 + 2^1 + 2^0 + 2^{-1} + 2^{-2}$

QUANTIFICATION DES SORTIES :  $-2^2 + 2^1 + 2^0$

## QUANTIFICATION DU SIGNAL II ET IQ



## TABLE DE VERITE

	0i2	0i1	0i0	oq2	oq1	oq0
11111	1	1	1	0	0	1
11110	1	1	1	0	0	1
11101	1	1	1	0	0	1
11100	1	1	1	0	0	1
11011	1	1	1	0	0	1
11010	1	1	0	0	1	0
11001	1	1	0	0	1	0
11000	1	1	0	0	1	0
10111	1	1	0	0	1	0
10110	1	1	0	0	1	0
10101	1	0	1	0	1	1
10100	1	0	1	0	1	1
10011	1	0	1	0	1	1
10010	1	0	1	0	1	1
10001	1	0	1	0	1	1
10000	1	0	1	0	1	1
01111	0	1	1	1	0	1
01110	0	1	1	1	0	1
01101	0	1	1	1	0	1
01100	0	1	1	1	0	1
01011	0	1	1	1	0	1
01010	0	1	0	1	1	0
01001	0	1	0	1	1	0
01000	0	1	0	1	1	0
00111	0	1	0	1	1	0
00110	0	1	0	1	1	0
00101	0	0	1	1	1	1
00100	0	0	1	1	1	1
00011	0	0	1	1	1	1
00010	0	0	1	1	1	1
00001	0	0	1	1	1	1
00000	0	0	1	1	1	1

LACINE

ECOLE DE TECHNOLOGIE SUPERIEURE

TITRE DU PROJET: NCLOS/QPSE SYSTEM

DESCRIPTION DE LA PAGE: QUANTIFICATION LINEAIRE

CODÉ DE PROJET: PRJ\_0001  
CODÉ D'OBJET: VT\_00010101

NOM: QUANT LINEAIRE  
DESSINÉ PAR: Mario Lauson  
DATE: 13/09/00

# QUANTIFICATION NON LINEAIRE DE I ET Q

QUANTIFICATION DES ENTREES II ET IQ :  $-2^2 + 2^1 + 2^0 + 2^{-1} + 2^{-2}$   
 QUANTIFICATION DES SORTIES :  $-2^2 + 2^1 + 2^0$

EXEMPLE DE QUANTIFICATION DU SIGNAL II (IDEM POUR IQ)

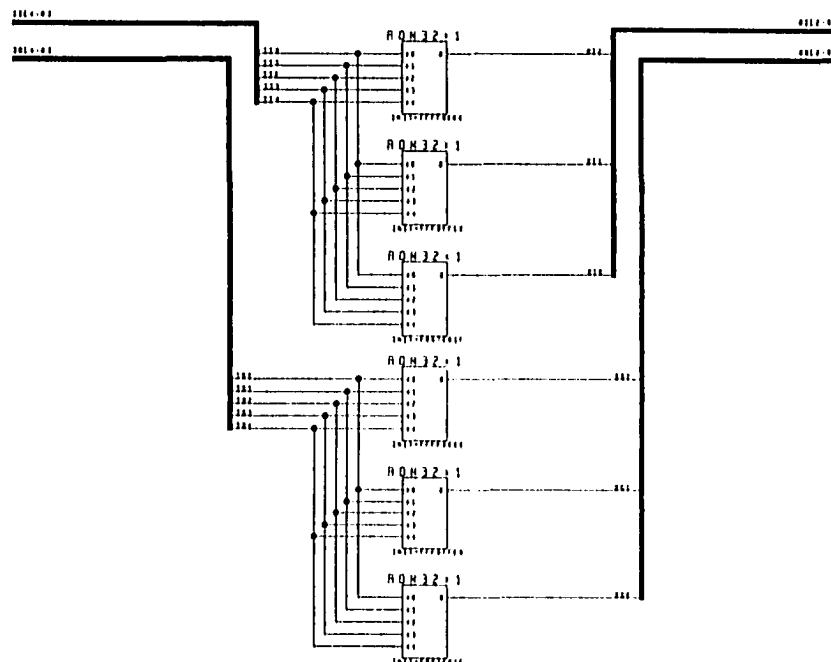
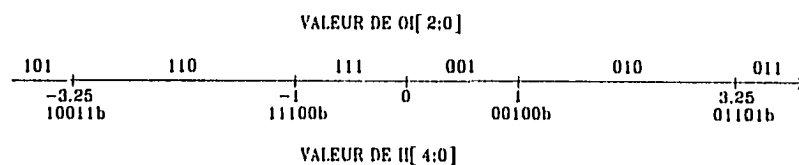


TABLE DE VERITE

	02	01	00
11111	1	1	1
11110	1	1	1
11101	1	1	1
11100	1	1	1
11011	1	1	0
11010	1	1	0
11001	1	1	0
11000	1	1	0
10111	1	1	0
10110	1	1	0
10101	1	1	0
10100	1	1	0
10011	1	1	0
10010	1	0	1
10001	1	0	1
10000	1	0	1
01111	0	1	1
01110	0	1	1
01101	0	1	0
01100	0	1	0
01011	0	1	0
01010	0	1	0
01001	0	1	0
01000	0	1	0
00111	0	1	0
00110	0	1	0
00101	0	1	0
00100	0	0	1
00011	0	0	1
00010	0	0	1
00001	0	0	1
00000	0	0	1

LACINE

ECOLE DE TECHNOLOGIE SUPERIEURE

TITRE DU PROJET HCLAS/OPSC SYSTEM

DESCRIPTION DE LA PAGE QUANTIFICATION LINEAIRE

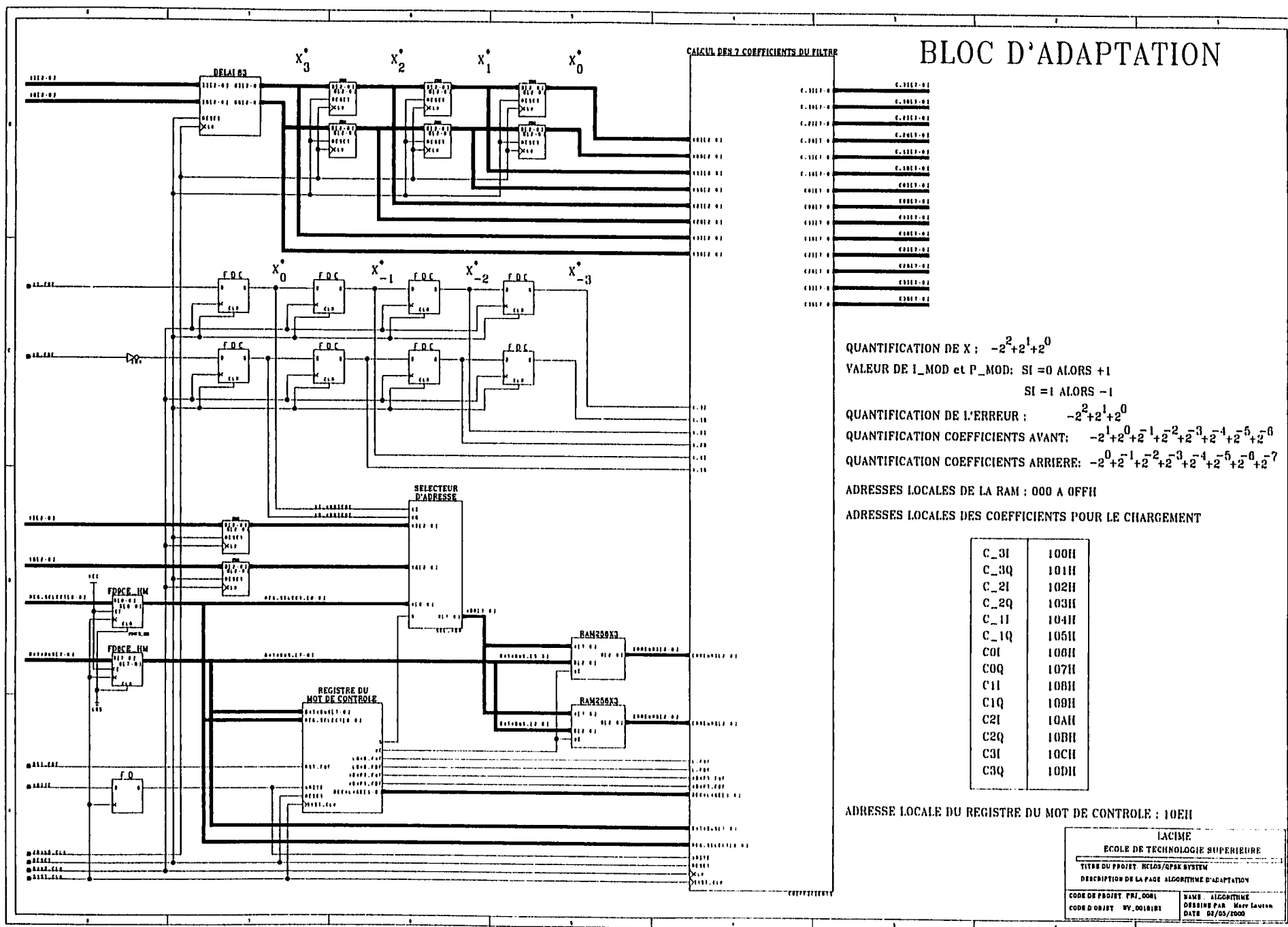
CODE DE PROJET PAJ\_D001

CODE D'OBJET NY\_001010

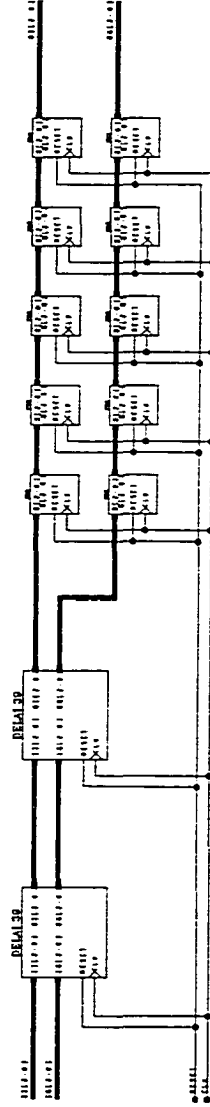
NOM : QUANT LINEAIRE

DESIGNE PAR Marc LAUSTON

DATE 10/05/00



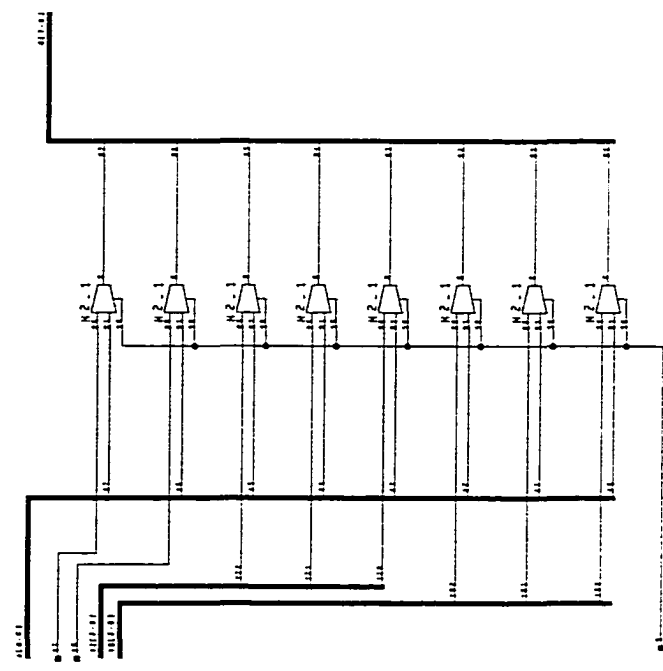
# DELAI DE 83 COUPS D'HORLOGE



<p>LACINE</p>	
<p>ECOLE DE TECHNOLOGIE SUPERIEURE</p>	
<p>TITRE DU PROJET: DELAI DE 83 COUPS D'HORLOGE</p>	
<p>DESCRIPTION DE LA PAGES: DELAI DE 83 COUPS D'HORLOGE</p>	
<p>CODER DU PROJET: PAI_0001</p>	<p>DATE: 11/07/2000</p>
<p>CODER DU PROJET: 01_0001</p>	<p>DATE: 11/07/2000</p>

# SELECTEUR D'ADRESSE

SI S=0 ALORS  $0[7:0] = X1 \ Xq \ Y1[2:0] \ Yq[2:0]$   
 SI S=1 ALORS  $0[7:0] = A[7:0]$



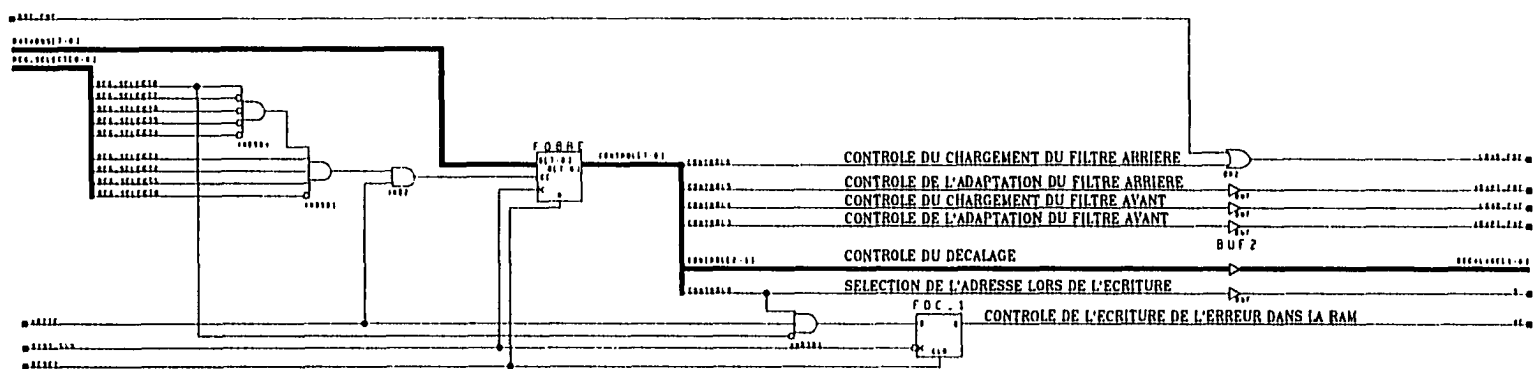
LACINE	
ECOLE DE TECHNOLOGIE SUPERIEURE	
TITRE DU PROJET ICLES/QUE SYSTEME	
DESCRIPTION DE LA PAGE SELECTEUR D'ADRESSE	
NOM : SELECTEUR	DATE : 10/05/2018
CODE DE PROJET PM, DOMI	CODE D'ORIENT BY, DOIRIAT



# REGISTRE DE CONTROLE

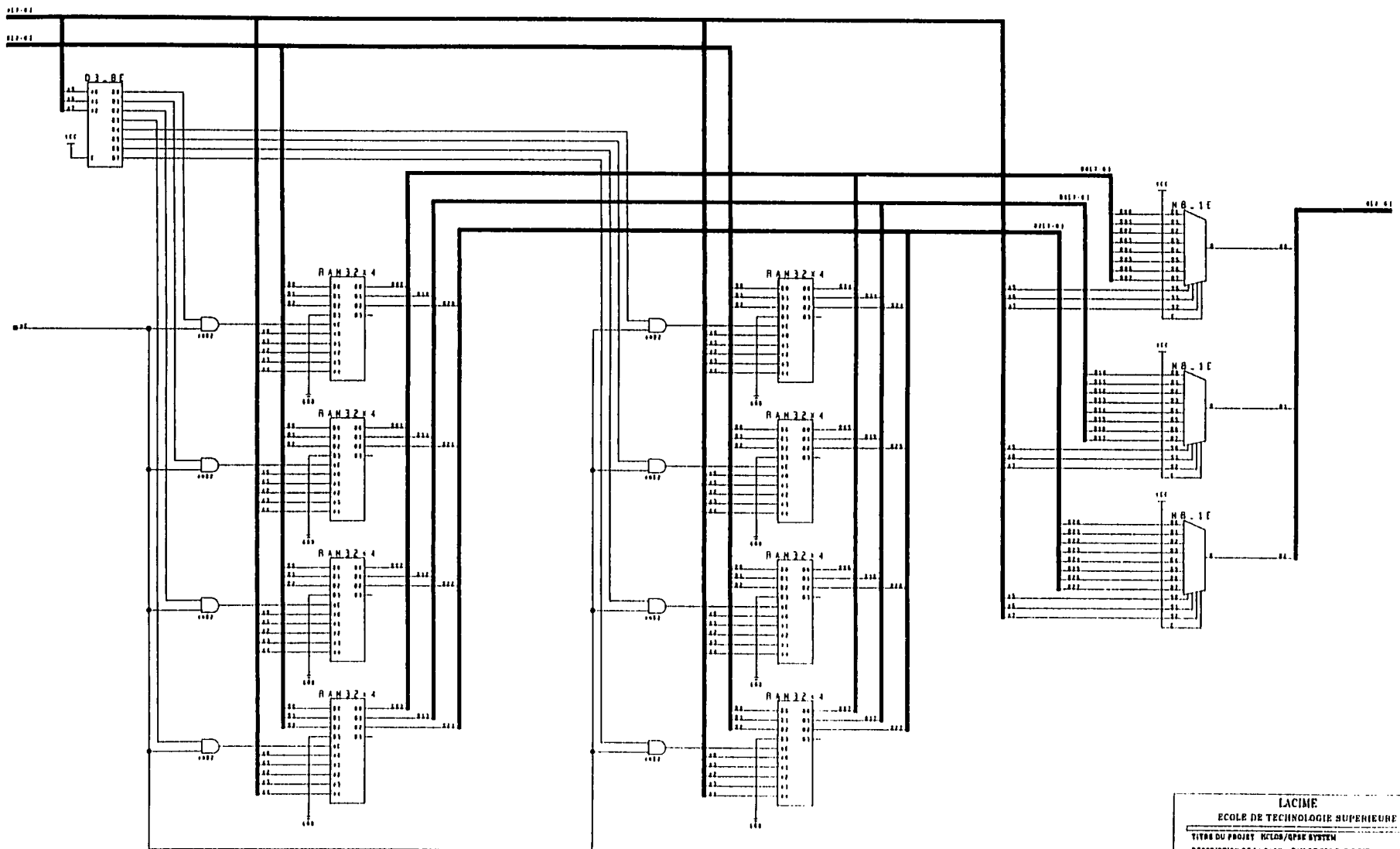
ADRESSE DU REGISTRE DU MOT DE CONTROLE : 10EH

NOT USE	LOAD FBF	ADAPT FBF	LOAD FWF	ADAPT FWF	DECALAGE1	DECALAGE0	WE
---------	----------	-----------	----------	-----------	-----------	-----------	----



LACIME	
ECOLE DE TECHNOLOGIE SUPERIEURE	
TITRE DU PROJET: MCIOS/QPSK SYSTEM	
DESCRIPTION DE LA PAGE: REGISTRE DE CONTROLE	
CODE DE PROJET: PRJ_0001	NOM: REGISTRE CONTROLE
CODE D'OBJET: RV_001010	DESSINE PAR: MARC LAUTHON
	DATE: 21/06/2000

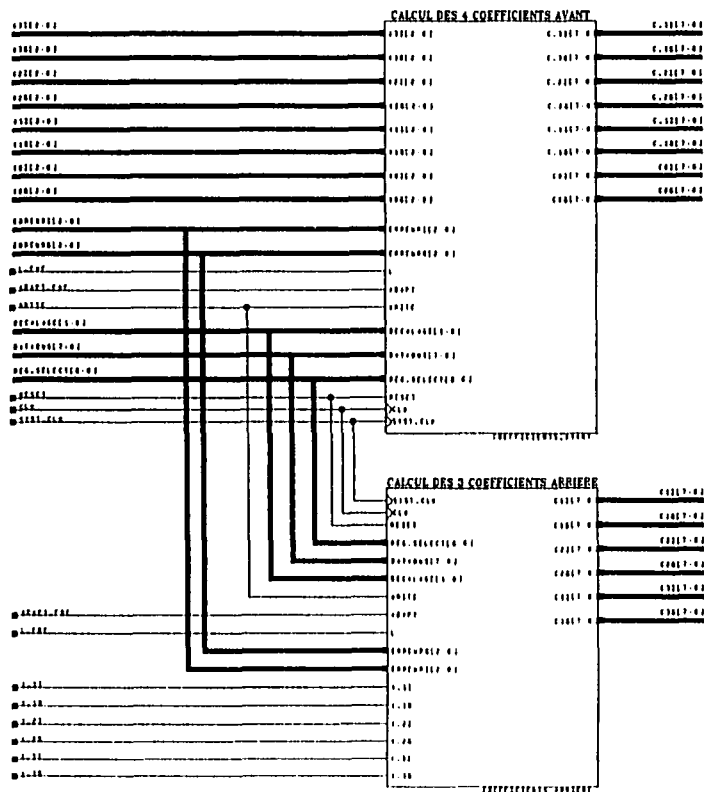
# MEMOIRE DE 256 PAR 3 BITS



LACINE	
ECOLE DE TECHNOLOGIE SUPERIEURE	
TITRE DU PROJET: MEMOIRE DE 256 PAR 3 BITS	
DESCRIPTION DE LA PAGE: RAM DE 256 PAR 3 BITS	
CODE DE PROJET: PRJ_0001	NOM: RAM256X3
CODE D'OBJET: WV_D01R1X	DESIGNE PAR: Marc Lanson
	DATE: 17/03/2000

## CALCUL DES 7 COEFFICIENTS COMPLEXES DE L'EGALISEUR

$$C_{+1} = C + X^* \text{erreur} \cdot \text{decalage}$$



QUANTIFICATION DE X POSITIF:  $-2^2 + 2^1 + 2^0$

VALEUR DE X NEGATIF: SI  $X=0$  ALORS +1

SI X=1 ALORS -1

QUANTIFICATION DE L'ERREUR :  $-2^2 + 2^1 + 2^0$

QUANTIFICATION COEFFICIENTS AVANT:  $-2^1 + 2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6}$

QUANTIFICATION COEFFICIENTS ARRIERE:  $-2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7}$

ADAPTATION DES COEFFICIENTS SI ADAPT = 1

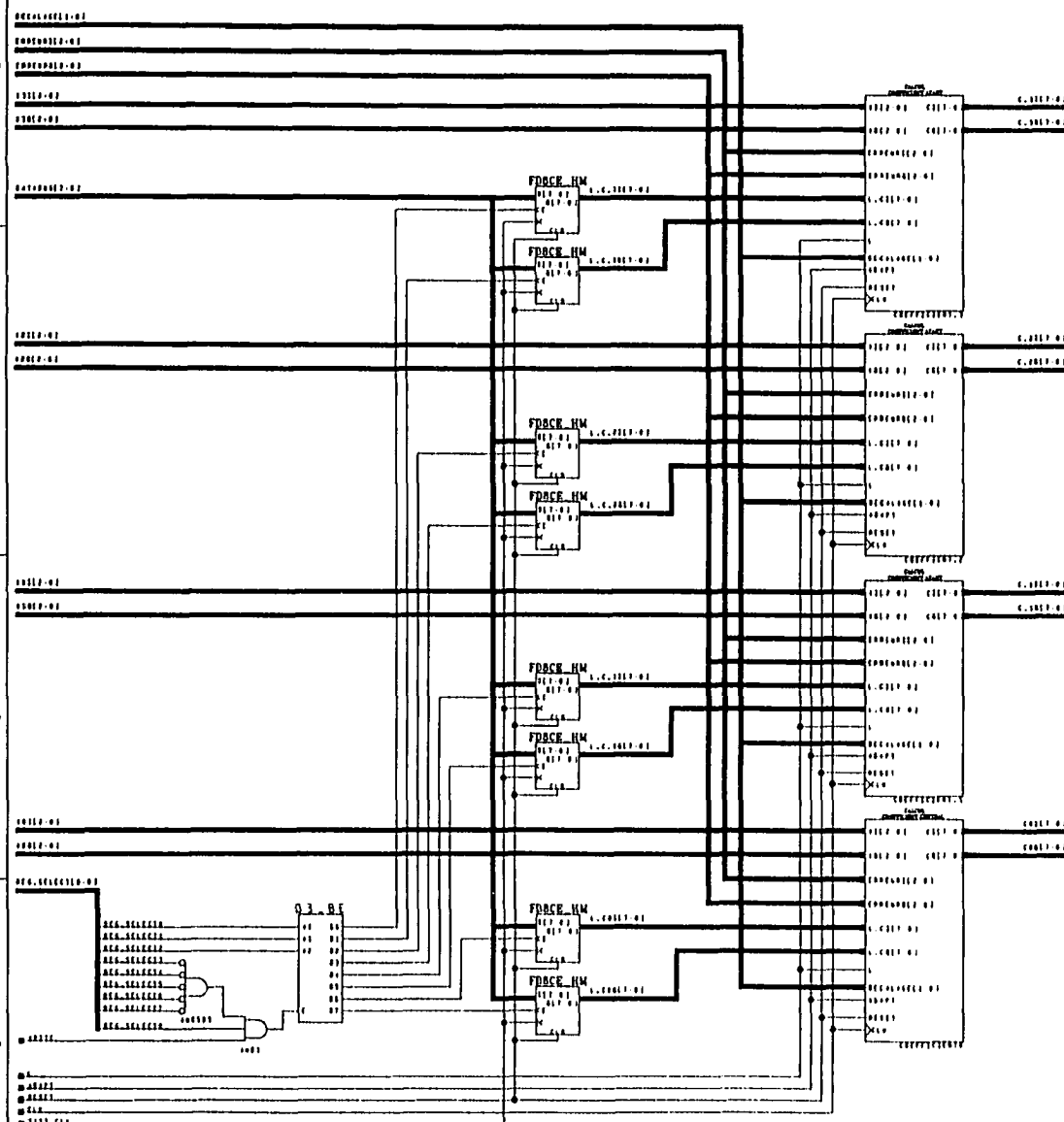
DECALAGE[ 1:0]	
0 0	$\times 10^{-3}$
0 1	$\times 10^{-4}$
1 0	$\times 10^{-5}$
1 1	$\times 10^{-6}$

### CHARGEMENT DES COEFFICIENTS SI $L = 1$

### ADRESSES LOCALES DES COEFFICIENTS POUR LE CHARGEMENT

C_3I	100H
C_3Q	101H
C_2I	102H
C_2Q	103H
C_1I	104H
C_1Q	105H
COI	106H
COQ	107H
CI1	108H
CIQ	109H
C2I	10AH
C2Q	10BH
C3I	10CH
C3Q	10DH

# CALCUL DES 4 COEFFICIENTS COMPLEXES DU FILTRE AVANT



QUANTIFICATION DE X :

$$-2^2 + 2^1 + 2^0$$

QUANTIFICATION DE L'ERREUR :

$$-2^2 + 2^1 + 2^0$$

QUANTIFICATION COEFFICIENT L\_C :

$$-2^1 + 2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6}$$

QUANTIFICATION SORTIE :

$$-2^1 + 2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6}$$

ADAPTATION DES COEFFICIENTS SI ADAPT = 1

DECALAGE[1:0]	
0 0	$\times 10^{-3}$
0 1	$\times 10^{-4}$
1 0	$\times 10^{-5}$
1 1	$\times 10^{-6}$

CHARGEMENT DES COEFFICIENTS L\_C SI L = 1

ADRESSES LOCALES DES COEFFICIENTS POUR LE CHARGEMENT

C_3I	100H
C_3Q	101H
C_2I	102H
C_2Q	103H
C_1I	104H
C_1Q	105H
COI	106H
COQ	107H

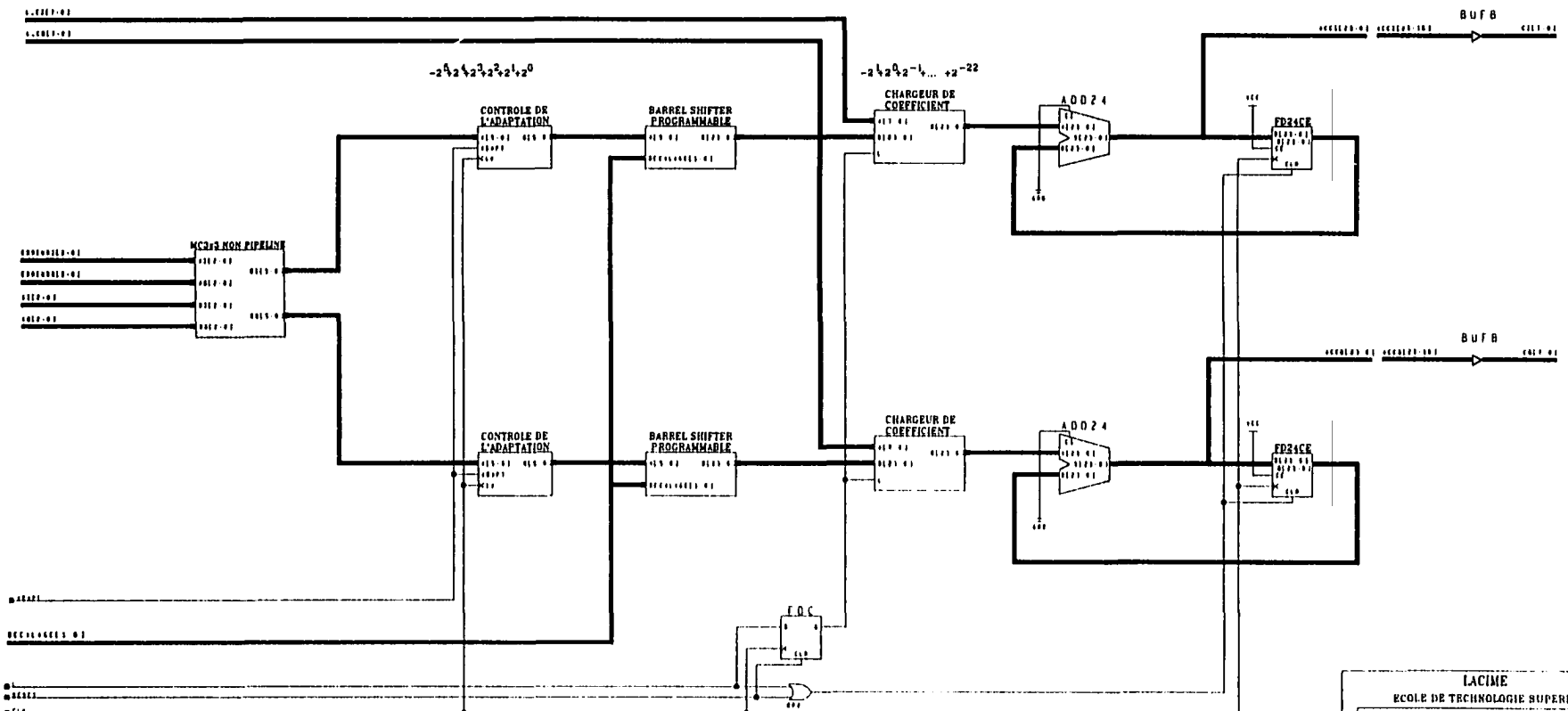
# CALCUL D'UN COEFFICIENT COMPLEXE DU FILTRE AVANT

QUANTIFICATION DE X :  $-2^2 + 2^1 + 2^0$   
 QUANTIFICATION DE L'ERREUR :  $-2^2 + 2^1 + 2^0$   
 QUANTIFICATION COEFFICIENT L\_C :  $-2^1 + 2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6}$   
 QUANTIFICATION SORTIE :  $-2^1 + 2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6}$

ADAPTATION DES COEFFICIENTS SI ADAPT = 1  
 CHARGEMENT DES COEFFICIENTS L\_C SI L = 1

DECALAGE[1:0]	
0 0	$\times 10^{-3}$
0 1	$\times 10^{-4}$
1 0	$\times 10^{-5}$
1 1	$\times 10^{-6}$

$$C_{+1} = C + X \cdot \text{erreur} \cdot \text{decalage} \quad \text{SI } L=0$$

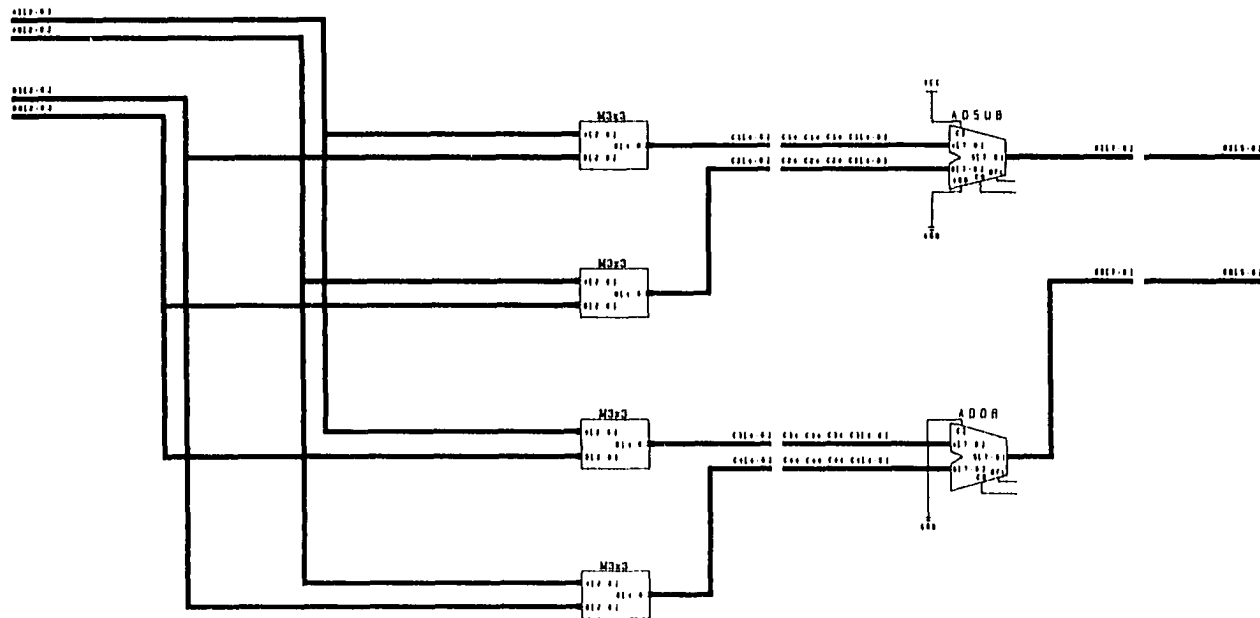


# MULTIPLICATEUR COMPLEXE DE 3 BITS PAR 3 BITS SANS PIPELINAGE

QUANTIFICATION ENTREE AI :  $-2^2 + 2^1 + 2^0$

QUANTIFICATION ENTREE BI :  $-2^2 + 2^1 + 2^0$

QUANTIFICATION DES SORTIES :  $-2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0$



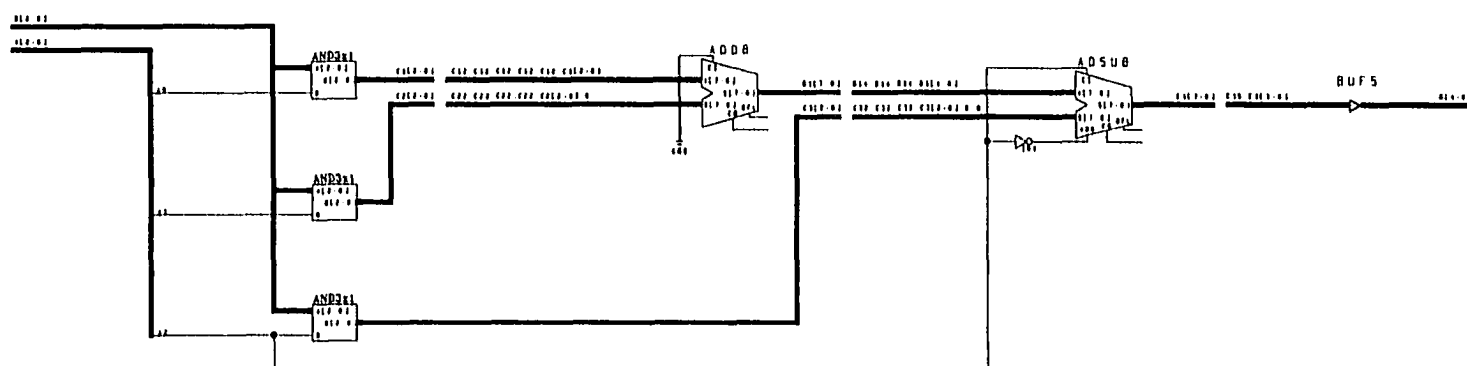
LACIME	
ECOLE DE TECHNOLOGIE SUPERIEURE	
TITRE DU PROJET: HCLAR/QPHE SYSTEM	
DESCRIPTION DE LA PAGE: MULTIPLICATEUR COMPLEXE	
CODR DE PROJET: PRJ_0001	NOM: MCS-2, NON PIPELINE
CODR D'OBJET: VY_00010101	DESSINE PAR: Marc Lachon
	DATE: 20/03/2000

# MULTIPLICATEUR 3 BITS PAR 3 BITS SANS PIPELINAGE

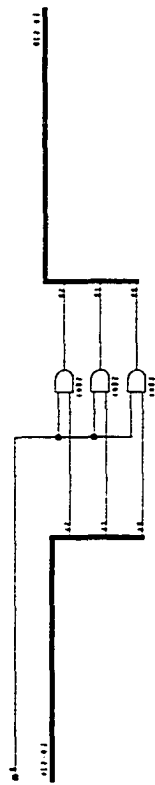
QUANTIFICATION ENTREE A :  $-2^2 + 2^1 + 2^0$

QUANTIFICATION ENTREE B :  $-2^2 + 2^1 + 2^0$

QUANTIFICATION DE SORTIE :  $-2^4 + 2^3 + 2^2 + 2^1 + 2^0$



# ET LOGIQUE DE 3 BITS PAR 1 BIT

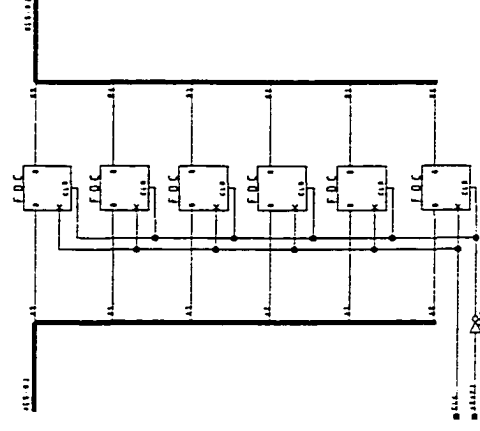


LACINE	
ECOLE DE TECHNOLOGIE SUPERIEURE	
TITRE DU PROJET REGIM/RTS SYSTEM	
DESCRIPTION DE LA PAGE ET LOGIQUE	
COURS DE PROJET PFI 5041	NOME ANDRIE
COURS DE PROJET PFI 5041	DESIGNE PAR Marc Lachin
COURS DE PROJET PFI 5041	DATE 20/01/2000



# CONTROLE DE L'ADAPTATION

SI ADAPT = 0 ALORS  $0[5:0] = 0$   
SI ADAPT = 1 ALORS  $0[5:0] = A[5:0]$

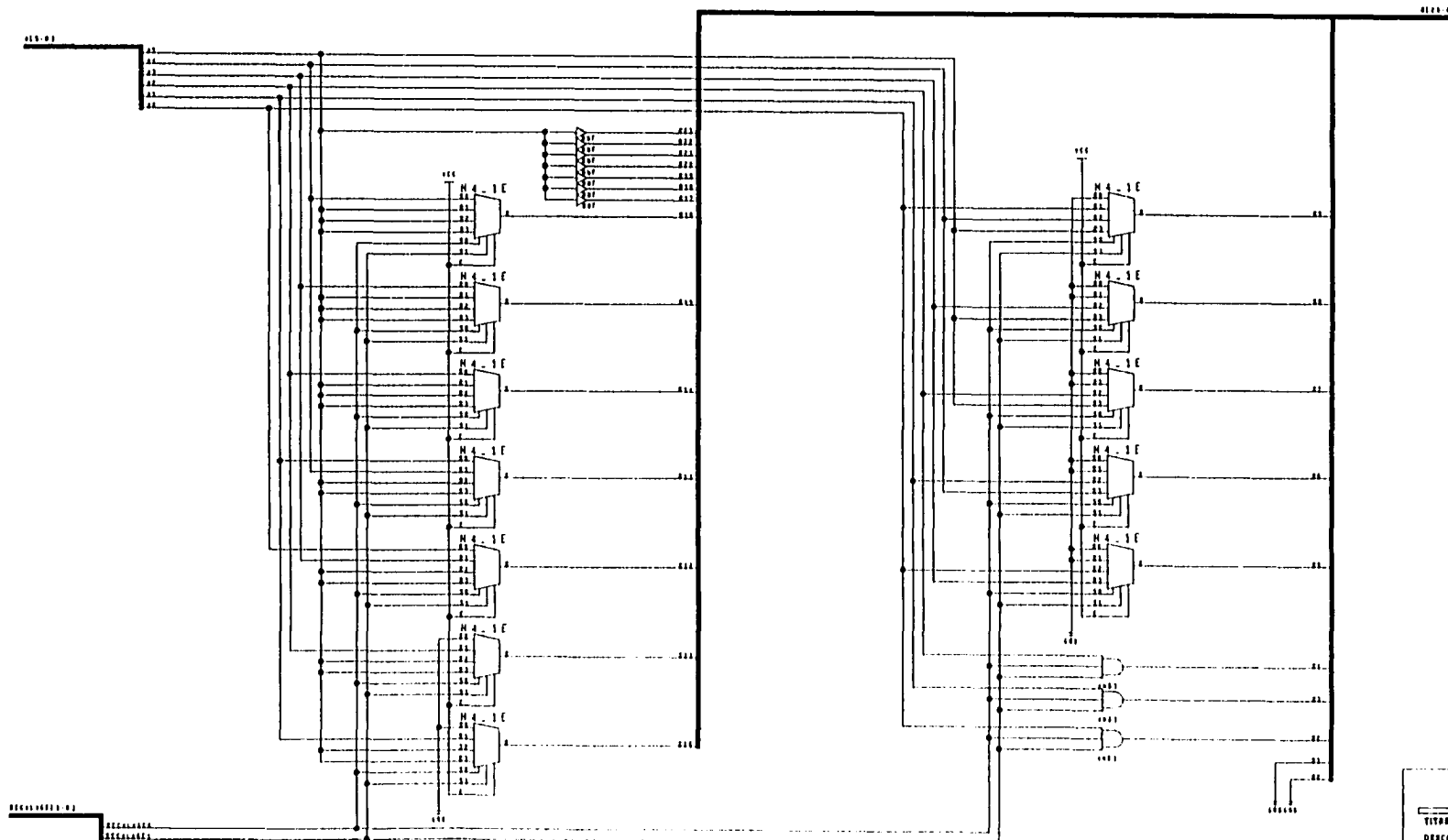


LACHE	
ECOLE DE TECHNOLOGIE SUPERIEURE	
TITRE DU PROJET: ECOS/CPH/STEN	
DESCRIPTION DE LA PAGE: CONTROLE DE L'ADAPTATION	
CODE DE PROJET: PHT_0001	NOM: CONTROLE ADAPTATION
CODE D'OBJET: PY_001010	DATE: 15/04/2006

# "BARREL SHIFTER" PROGRAMMABLE AVEC EXTENSION DE SIGNE

QUANTIFICATION ENTREE :  $-2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0$   
 QUANTIFICATION SORTIE :  $-2^1 + 2^0 + 2^{-1} + \dots + 2^{-22}$

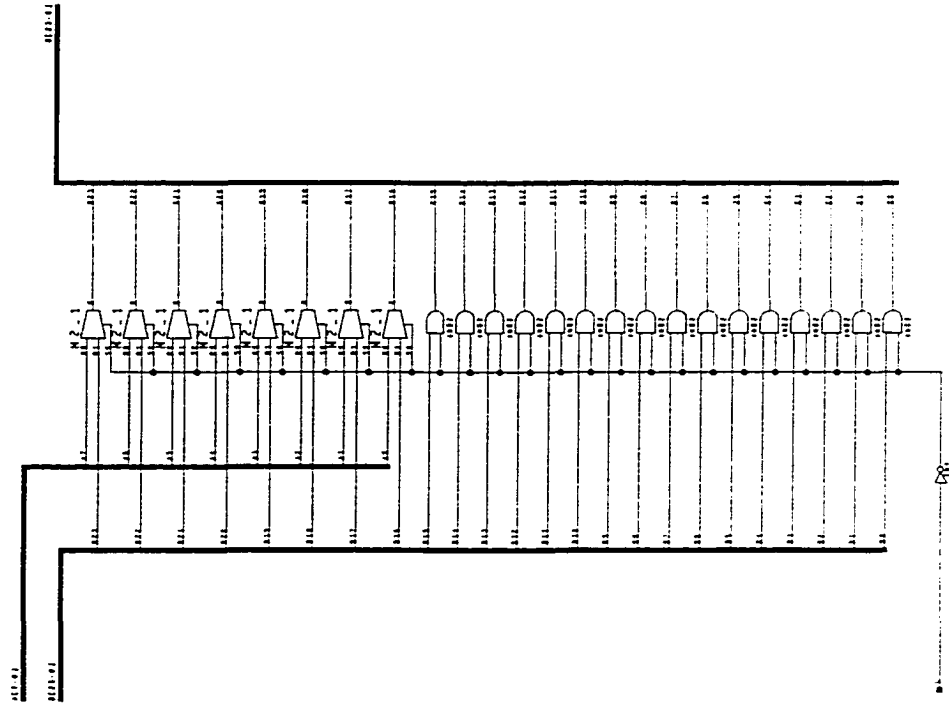
DECALAGE[1:0]	O[23:0]	
0 0	A5 A5 A5 A5 A5 A5 A5 A4 A3 A2 A1 A0 000000000000	x 10 <sup>-3</sup>
0 1	A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A4 A3 A2 A1 A0 0000000000	x 10 <sup>-4</sup>
1 0	A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A4 A3 A2 A1 A0 00000	x 10 <sup>-5</sup>
1 1	A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A4 A3 A2 A1 A0 00	x 10 <sup>-6</sup>



LACIME	
ECOLE DE TECHNOLOGIE SUPERIEURE	
TITRE DU PROJET: DECALAGE/QUANTIFICATION SYSTEM	
DESCRIPTION DE LA PAGE: "BARREL SHIFTER" PROG	
CODE DE PROJET: PRJ_0001	NOM: BARREL_SHIFTER
CODE D'OBJET: WY_0010101	DESIGNE PAR: WY_0010101
	DATE: 12/04/2000

# CHARGEUR DE COEFFICIENT

SI L = 0 ALORS 0[23:0] = n[23:0]  
 SI L = 1 ALORS 0[23:0] = A[7:0] 00000000000000000000



LACINE	
ECOLE DE TECHNOLOGIE SUPERIEURE	
TITRE DU PROJET RECU/QUPE ESTEN	
DESCRIPTION DE LA PAGE FEUILLE DE COEFFICIENT	
CODS DU PROJET P01 001	NOM CHARGEUR COEFFICIENT
CODS DU PROJET P1 001010	DESINE PAR M. LACINE
	DATE 11/03/2000

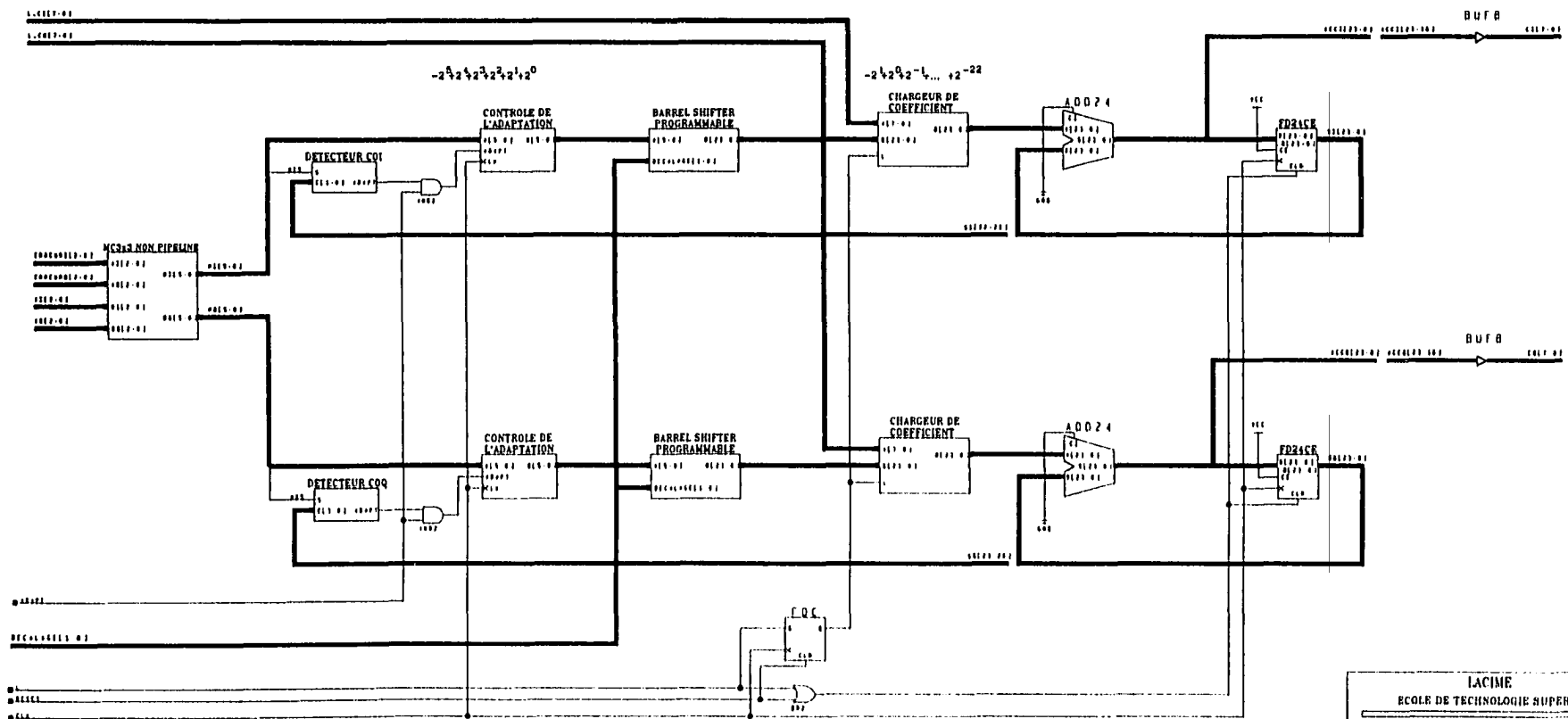
# CALCUL DU COEFFICIENT CENTRAL COMPLEXE

QUANTIFICATION DE X :  $-2^2 + 2^1 + 2^0$   
 QUANTIFICATION DE L'ERREUR :  $-2^2 + 2^1 + 2^0$   
 QUANTIFICATION COEFFICIENT L\_C :  $-2^1 + 2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6}$   
 QUANTIFICATION SORTIE :  $-2^1 + 2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6}$

ADAPTATION DES COEFFICIENTS SI ADAPT = 1  
 CHARGEMENT DES COEFFICIENTS L\_C SI L = 1

DECALAGE[1:0]	
0 0	$\times 10^{-3}$
0 1	$\times 10^{-4}$
1 0	$\times 10^{-5}$
1 1	$\times 10^{-6}$

$$C_{+1} = C + X \cdot \text{erreur} \cdot \text{decalage} \quad \text{SI } L=0$$



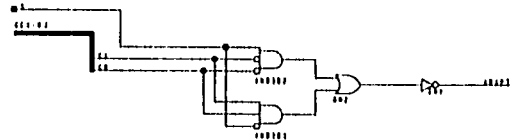
# DETECTEUR DE DEPASSEMENT POUR LA PARTIE REELLE DE C0

QUANTIFICATION DE C :  $-2^0 + 2^{-1}$

SI C < 0.5 et S = 1 ALORS ADAPT = 0

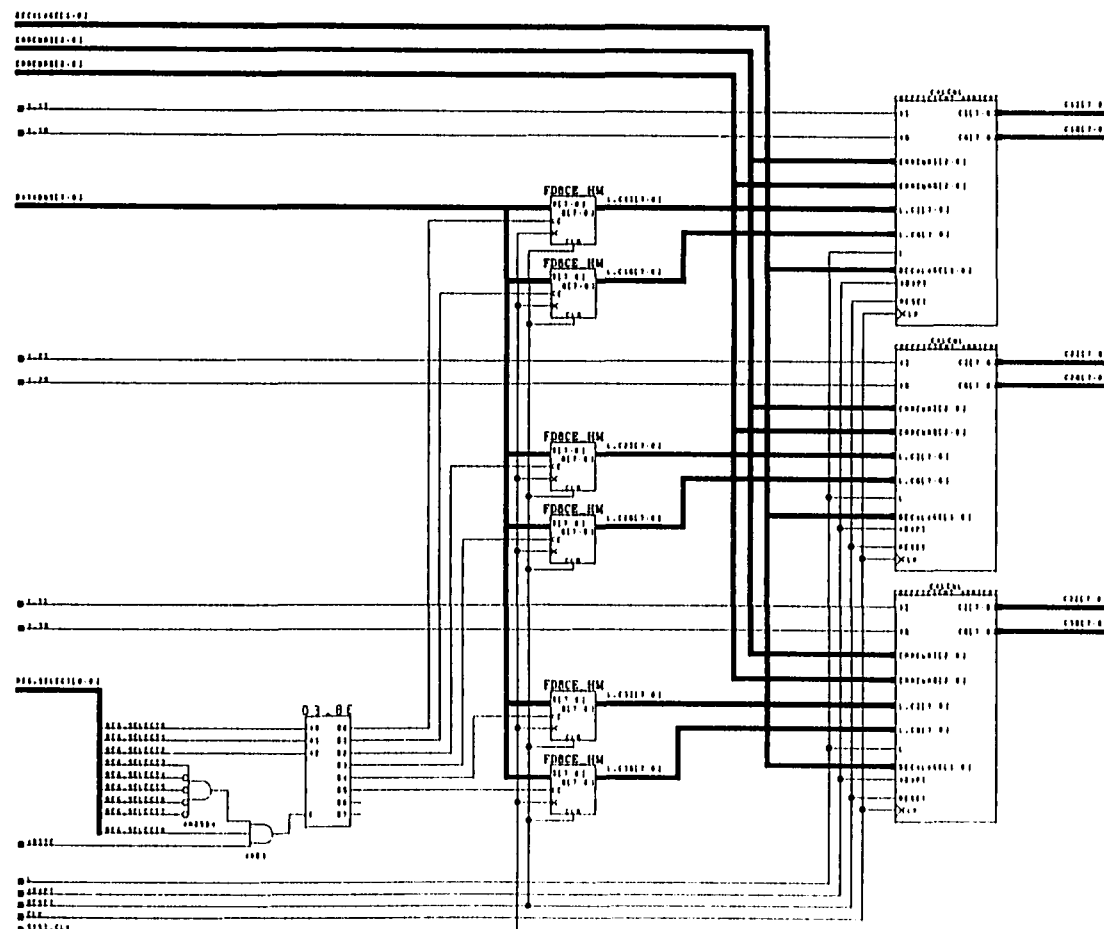
SI C >= 1.5 et S = 0 ALORS ADAPT = 0

SINON ADAPT = 1



LACINE	
ECOLE DE TECHNOLOGIE SUPERIEURE	
TITRE DU PROJET HCELOS/QPSK SYSTEM	
DESCRIPTION DE LA PAGE DETECTEUR DE DEPASSEMENT	
CODE DE PROJET PAJ_0001	NOM: DETECTEUR, C0
CODE D OBJET VV_0010191	DESSINE P.B. Marc Lachin
	DATE 23/09/2000

## CALCUL DES 3 COEFFICIENTS COMPLEXES DU FILTRE ARRIERE



VALEUR DE X : SI  $X=0$  ALORS +1

SI X=1 ALORS -1

QUANTIFICATION DE L'ERREUR :  $-2^2 + 2^1 + 2^0$

QUANTIFICATION COEFFICIENT L\_C :  $-2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7}$

QUANTIFICATION SORTIE :  ${}_{-2}^0 {}_{+2}^{-1} {}_{+2}^{-2} {}_{+2}^{-3} {}_{+2}^{-4} {}_{+2}^{-5} {}_{+2}^{-6} {}_{+2}^{-7}$

ADAPTATION DES COEFFICIENTS SI ADAPT = 1

DECALAGE[ 1:0]	
0 0	$\times 10^{-3}$
0 1	$\times 10^{-4}$
1 0	$\times 10^{-5}$
1 1	$\times 10^{-6}$

CHARGEMENT DES COEFFICIENTS L\_C SI L = 1

### ADRESSES LOCALES DES COEFFICIENTS POUR LE CHARGEMENT

C11	108H
C1Q	109H
C2I	10AH
C2Q	10BH
C3I	10CH
C3Q	10DH

# CALCUL D'UN COEFFICIENT COMPLEXE DU FILTRE ARRIERE

VALEUR DE X : SI X=0 ALORS +1

SI X=1 ALORS -1

QUANTIFICATION DE L'ERREUR :  $-2^2 + 2^1 + 2^0$

QUANTIFICATION COEFFICIENT L\_C :  $-2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7}$

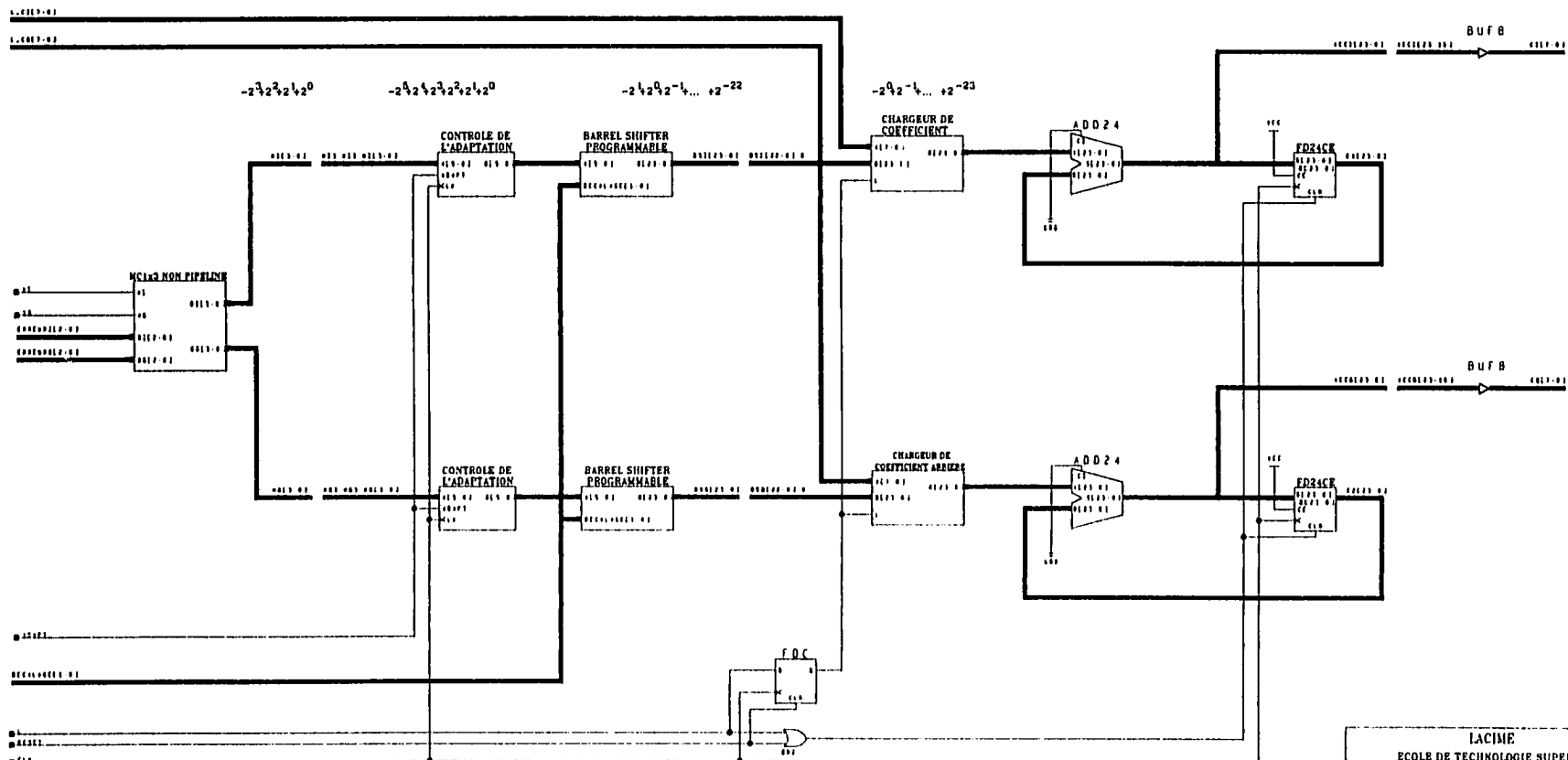
QUANTIFICATION SORTIE :  $-2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7}$

ADAPTATION DES COEFFICIENTS SI ADAPT = 1

CHARGEMENT DES COEFFICIENTS L\_C SI L = 1

DECALAGE[1:0]	
0 0	$\times 10^{-3}$
0 1	$\times 10^{-4}$
1 0	$\times 10^{-5}$
1 1	$\times 10^{-6}$

$$C_{+1} = C + X \cdot \text{erreur} \cdot \text{decalage} \quad \text{SI } L=0$$



LACINE	
ECOLE DE TECHNOLOGIE SUPERIEURE	
TITRE DU PROJET: NCLOS/QPSE SYSTÈM	
DESCRIPTION DE LA PAGE: CALCUL D'UN COEFFICIENT	
CODÉ DE PROJET: PRJ_0001	NOM: COEFFICIENT ARRIERE
CODÉ D'OBJET: VY_0001010	DESSINÉ PAR: Marc Lousen
	DATE: 18/08/2000

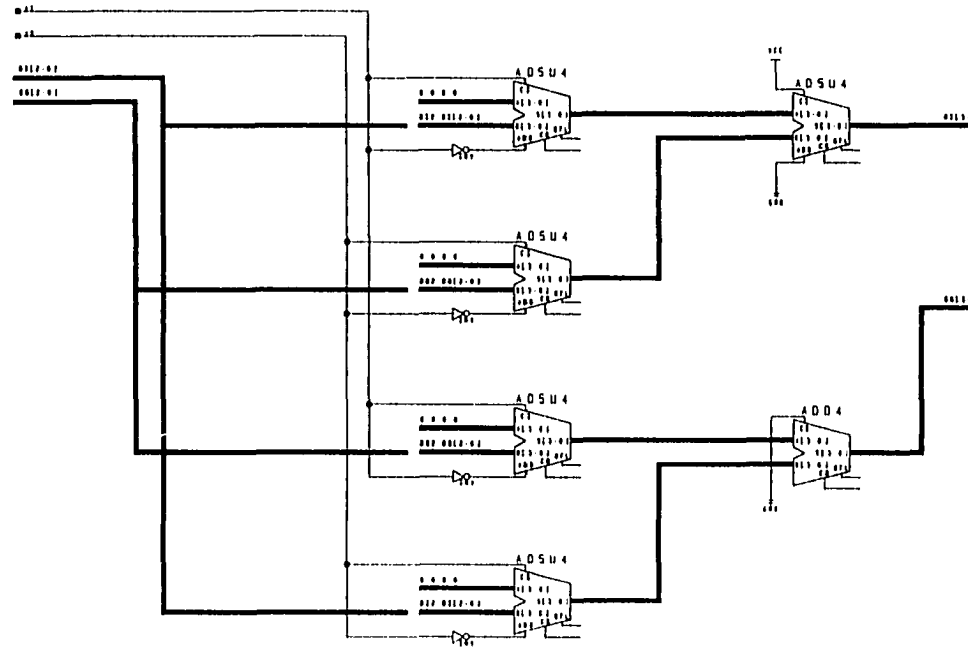
# MULTIPLICATEUR COMPLEXE DE 1 BIT (+/- 1) PAR 3 BITS

SI A=0 ALORS MULTIPLICATION PAR +1

SI A=1 ALORS MULTIPLICATION PAR -1

QUANTIFICATION DES ENTREES B :  $-2^2 + 2^1 + 2^0$

QUANTIFICATION DES SORTIES :  $-2^3 + 2^2 + 2^1 + 2^0$



LACINE	
ECOLE DE TECHNOLOGIE SUPERIEURE	
TITRE DU PROJET : MC103/OPSE SYSTEM	
DESCRIPTION DE LA PAGE : MULTIPLICATEUR COMPLEXE	
CODE DE PROJET : PM, 0001	NOM : MC103, NON, PIPELINE
CODE D'OBJET : PV, 0010102	DESSINE PAR : MATH LAMBERT
	DATE : 06/06/2000

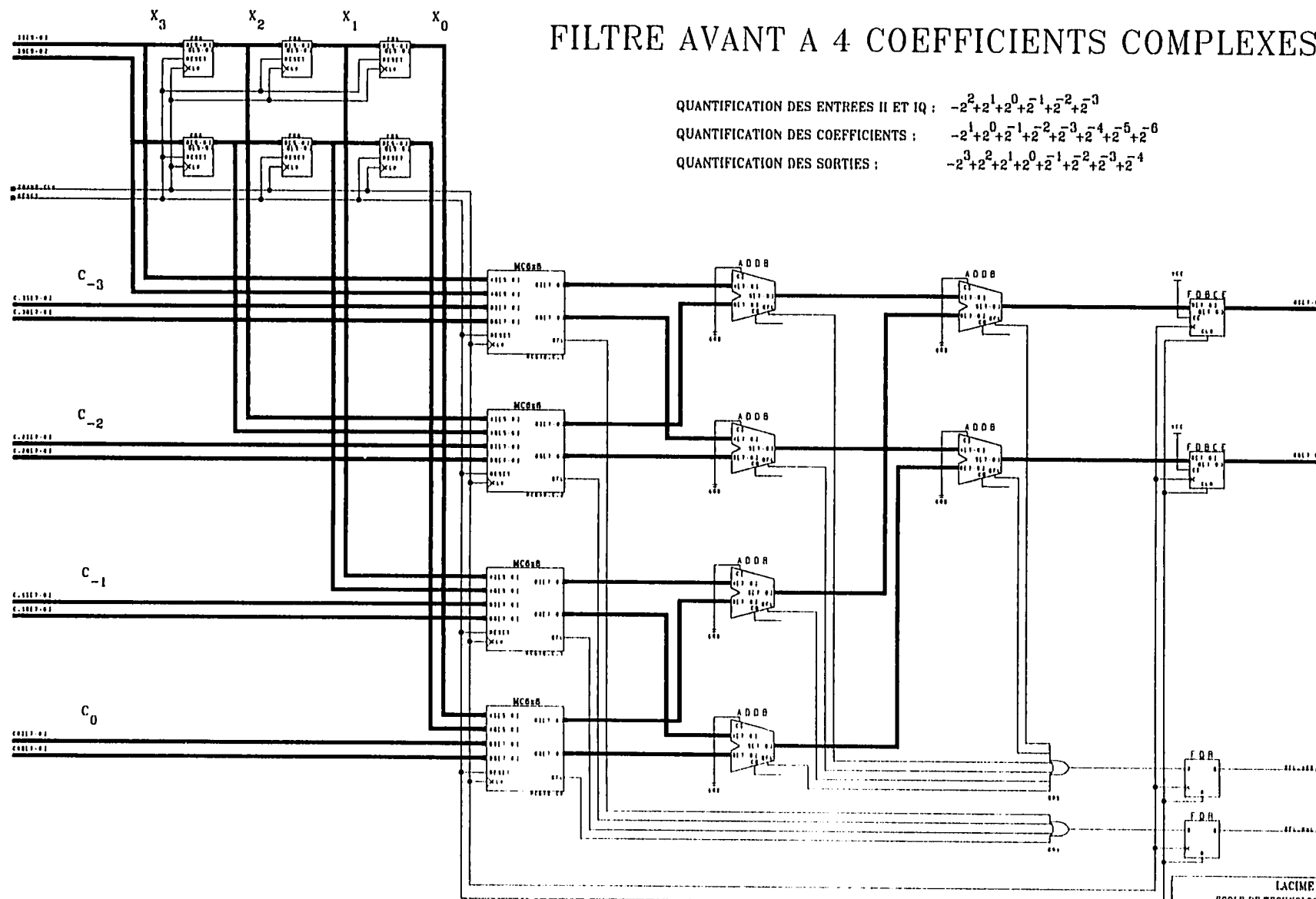


# FILTRE AVANT A 4 COEFFICIENTS COMPLEXES

QUANTIFICATION DES ENTREES II ET IQ :  $-2^2+2^1+2^0+2^{-1}+2^{-2}+2^{-3}$

QUANTIFICATION DES COEFFICIENTS :  $-2^1+2^0+2^{-1}+2^{-2}+2^{-3}+2^{-4}+2^{-5}+2^{-6}$

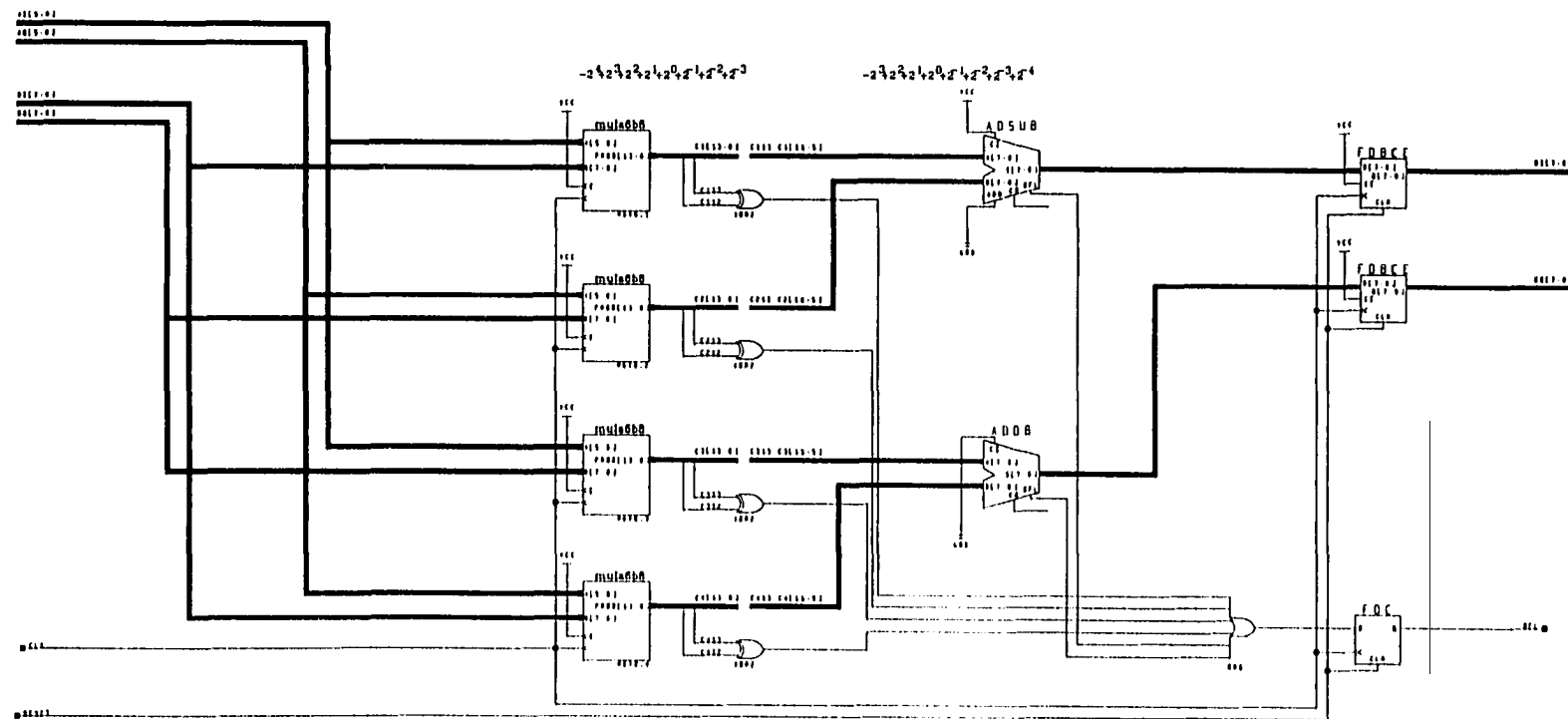
QUANTIFICATION DES SORTIES :  $-2^3+2^2+2^1+2^0+2^{-1}+2^{-2}+2^{-3}+2^{-4}$



LACINE	
ECOLE DE TECHNOLOGIE SUPERIEURE	
TITRE DU PROJET: MCLOR/SPR SYSTEM	
DESCRIPTION DE LA PAGE: FILTRE AVANT	
CODE DE PROJET: PAJ_0001	NOM: FILTREAV
CODE D'OBJET: WV_0010101	DESIGNED PAR: Marc LACINE
	DATE: 03/02/2000

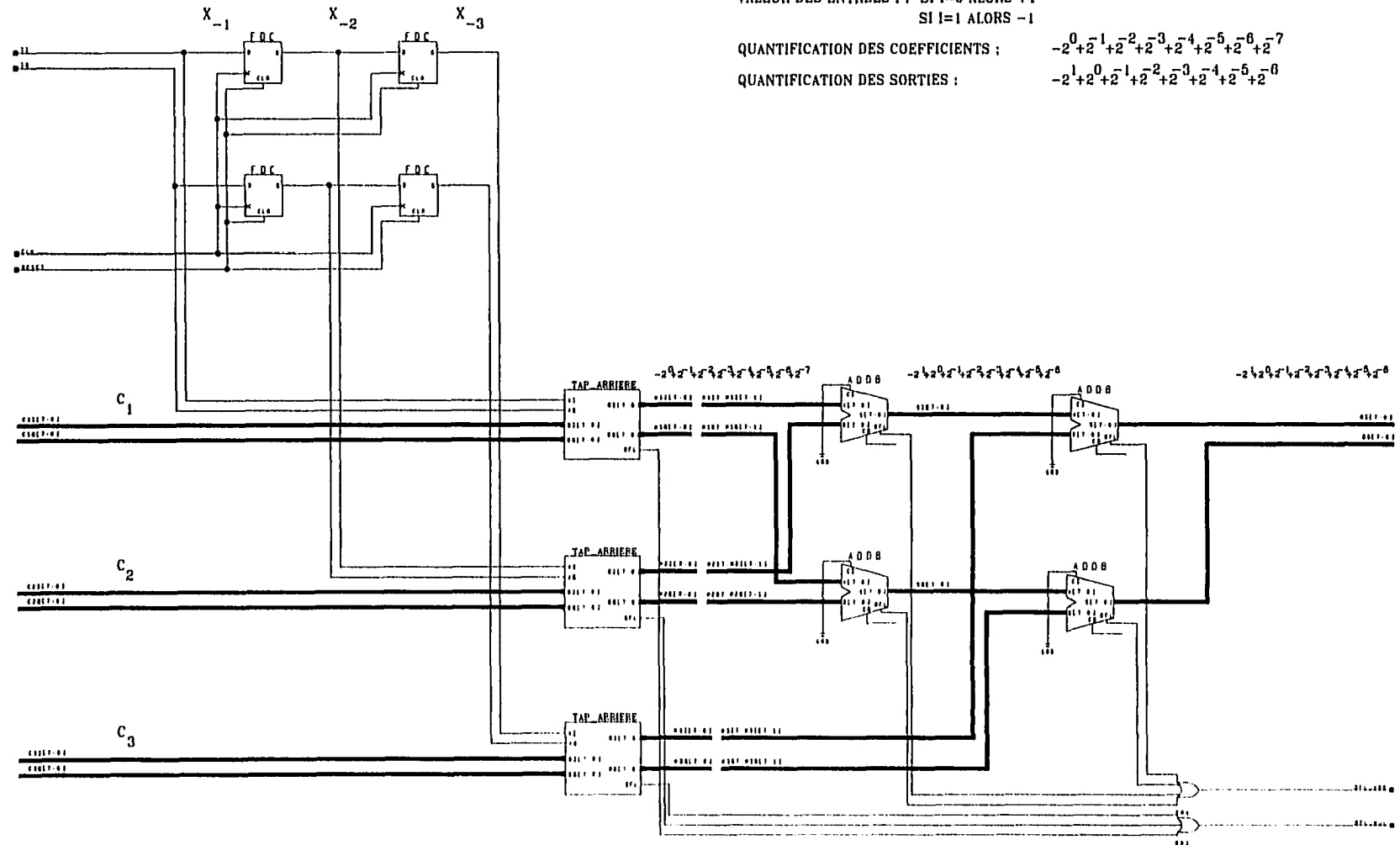
# MULTIPLICATEUR COMPLEXE DE 6 BITS PAR 8 BITS

QUANTIFICATION DES ENTREES A :  $-2^2 + 2^1 + 2^0 + 2^{-1} + 2^{-2} + 2^{-3}$   
 QUANTIFICATION DES ENTREES B :  $-2^1 + 2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6}$   
 QUANTIFICATION DES SORTIES :  $-2^3 + 2^2 + 2^1 + 2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4}$



LACIME	
ECOLE DE TECHNOLOGIE SUPERIEURE	
TITRE DU PROJET : MCD8/QPSK SYSTEM	
DESCRIPTION DE LA PAGE : MULTIPLICATEUR COMPLEXE	
CODE DE PROJET : PRJ_0061	NOM : MC68
CODE D'OBJET : TV_001818X	DESSINE PAR : Marc Janson
	DATE : 03/08/2000

# FILTRE ARRIERE A 3 COEFFICIENTS COMPLEXES



VALEUR DES ENTREES I : SI I=0 ALORS +1  
SI I=1 ALORS -1

QUANTIFICATION DES COEFFICIENTS :  $-2^0+2^{-1}+2^{-2}+2^{-3}+2^{-4}+2^{-5}+2^{-6}+2^{-7}$

QUANTIFICATION DES SORTIES :  $-2^1+2^0+2^{-1}+2^{-2}+2^{-3}+2^{-4}+2^{-5}+2^{-6}$

LACINE	
ECOLE DE TECHNOLOGIE SUPERIEURE	
TITRE DU PROJET NCDS/QPSK SYSTEM	
DESCRIPTION DE LA PAGE FILTRE ARRIERE	
CODE DE PROJET PRI_0001	NOM : FILTREAR
CODE D'OBJET NY_001010	DESIGNED PAR Marc Lachena
	DATE 01/02/2000

# MULTIPLICATEUR 8X1 COMPLEXE (TAP ARRIERE)

SI A=0 ALORS MULTIPLICATION PAR +1

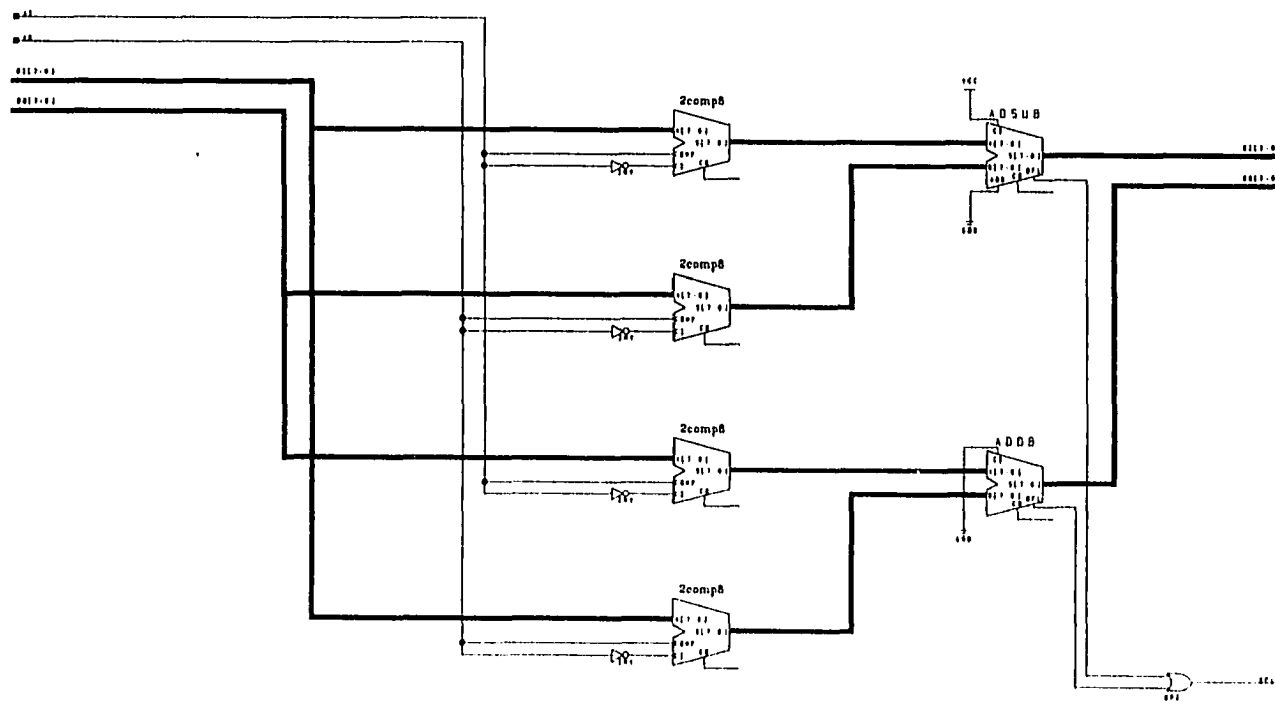
SI A=1 ALORS MULTIPLICATION PAR -1

QUANTIFICATION DES ENTREES B :  $-2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7}$

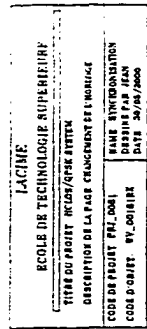
QUANTIFICATION DES SORTIES :  $-2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7}$

OI = (BI x AI) - (BQ x AQ)

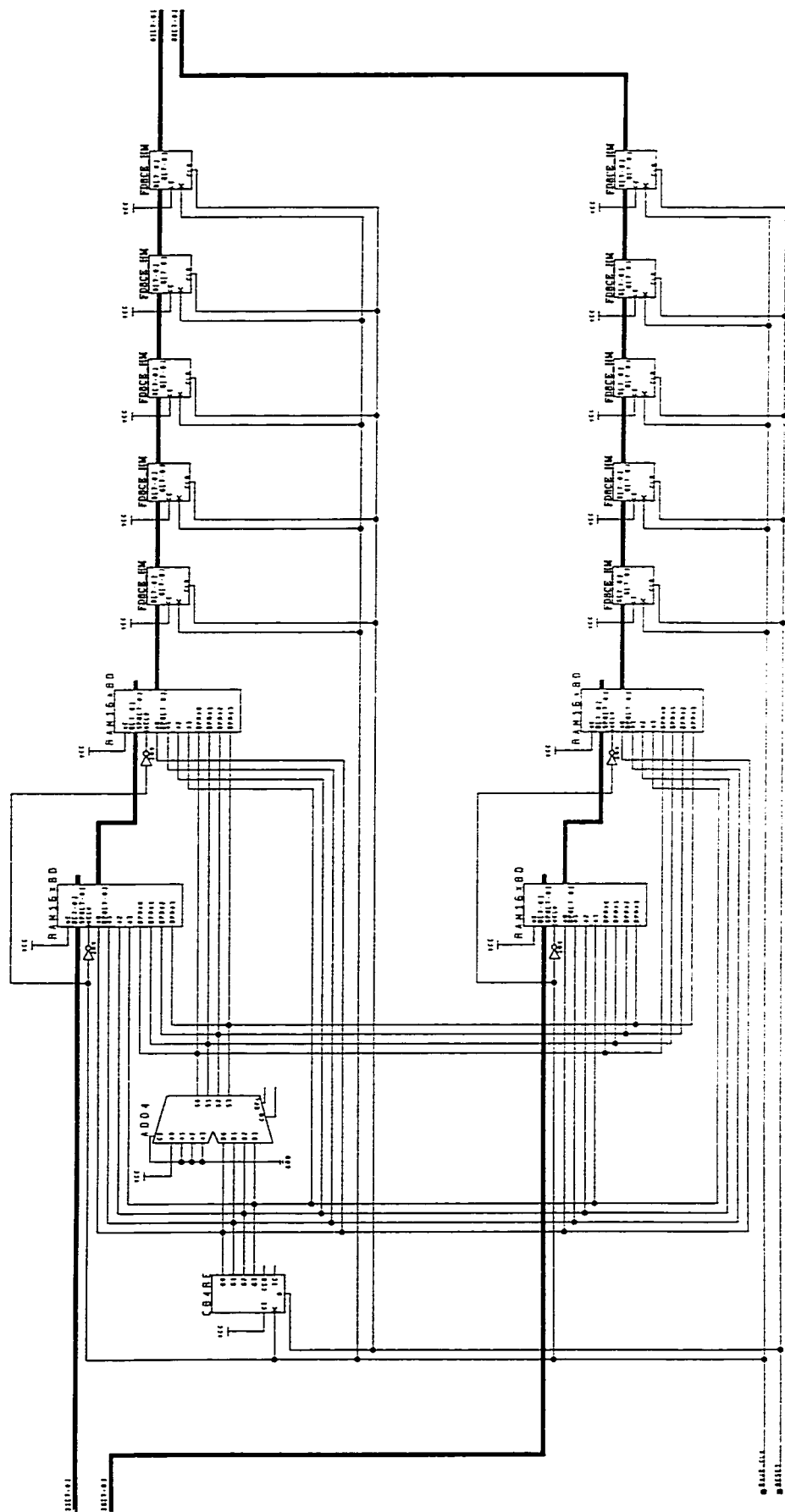
OQ = (BQ x AI) + (BI x AQ)



LACIME	
ECOLE DE TECHNOLOGIE SUPERIEURE	
TITRE DU PROJET HCIDS/QPSE SYSTEM	
DESCRIPTION DE LA PAGE TAP POUR LE FILTRE ARRIERE	
CODE DE PROJET PRJ_0001	NOM: TAP_ARRIERE
CODE D'OBJET: NY_0010101	DESIGNE PAR: MATH LAUREN
	DATE: 18/04/2000

[illegible]

# DELAI DE 35 COUPS D'HORLOGE



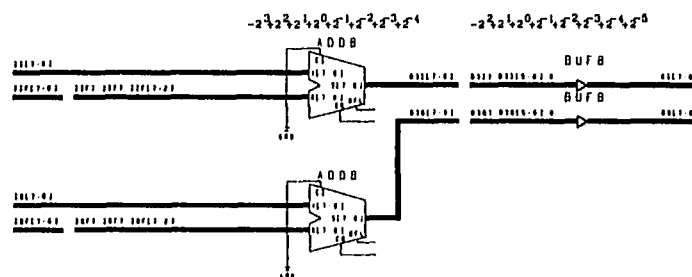
LACINE	
ECOLE DE TECHNOLOGIE SUPERIEUR	
TITRE DU PROJET RECO/SPR SYSTEM	
DESCRIPTION DE LA PAGE DELAI DE 35 COUPS D'HORLOGE	
COD DE PROJET PR_0001	PAGE DELAI3
COD D'OBJET NY_000000	DATE 21/02/2005

# BLOC DE SOMMATION COMPLEXE

QUANTIFICATION DES ENTREES II ET IQ :  $-2^3+2^2+2^1+2^0-1-2^{-1}-2^{-2}-2^{-3}-2^{-4}$

QUANTIFICATION DES ENTREES IIF ET IQF :  $-2^1+2^0-2^{-1}-2^{-2}-2^{-3}-2^{-4}+2^{-5}+2^{-6}$

QUANTIFICATION DES SORTIES :  $-2^2+2^1+2^0-2^{-1}-2^{-2}-2^{-3}-2^{-4}-2^{-5}$



LACINE	
ECOLE DE TECHNOLOGIE SUPERIEURE	
TITRE DU PROJET	HCIDS/QPSK SYSTEM
DESCRIPTION DE LA PAGE	BLOC DE SOMMATION COMPLEXE
CODE DE PROJET	PRJ_0001
CODE D'OBJET	WV_0010/RE
NOM : BONNE	
DESSINE PAR : Marc Louton	
DATE	11/02/2000

# FILTRE COMPLEXE A REPONSE IMPULSIONNELLE INFINIE A UN COEFFICIENT

VALEUR DE OI7 et OQ7 : SI =0 ALORS +1  
SI =1 ALORS -1

QUANTIFICATION DES ENTREES II ET IQ :

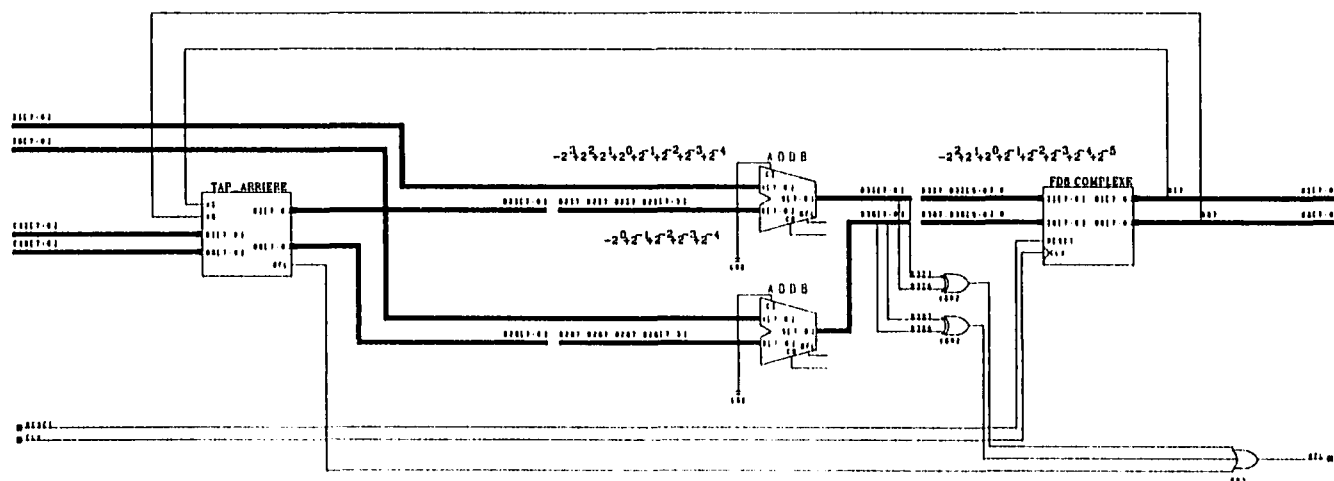
$$-2^3 + 2^2 + 2^1 + 2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4}$$

QUANTIFICATION DES ENTREES CII ET CIQ :

$$-2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7}$$

QUANTIFICATION DES SORTIES :

$$-2^2 + 2^1 + 2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5}$$





# QUANTIFICATEUR

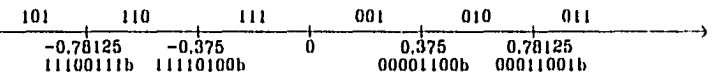
EXEMPLE DE QUANTIFICATION DU SIGNAL II (IDEM POUR IQ)

QUANTIFICATION DES ENTREES :  $-2^2 + 2^1 + 2^0 + 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5}$

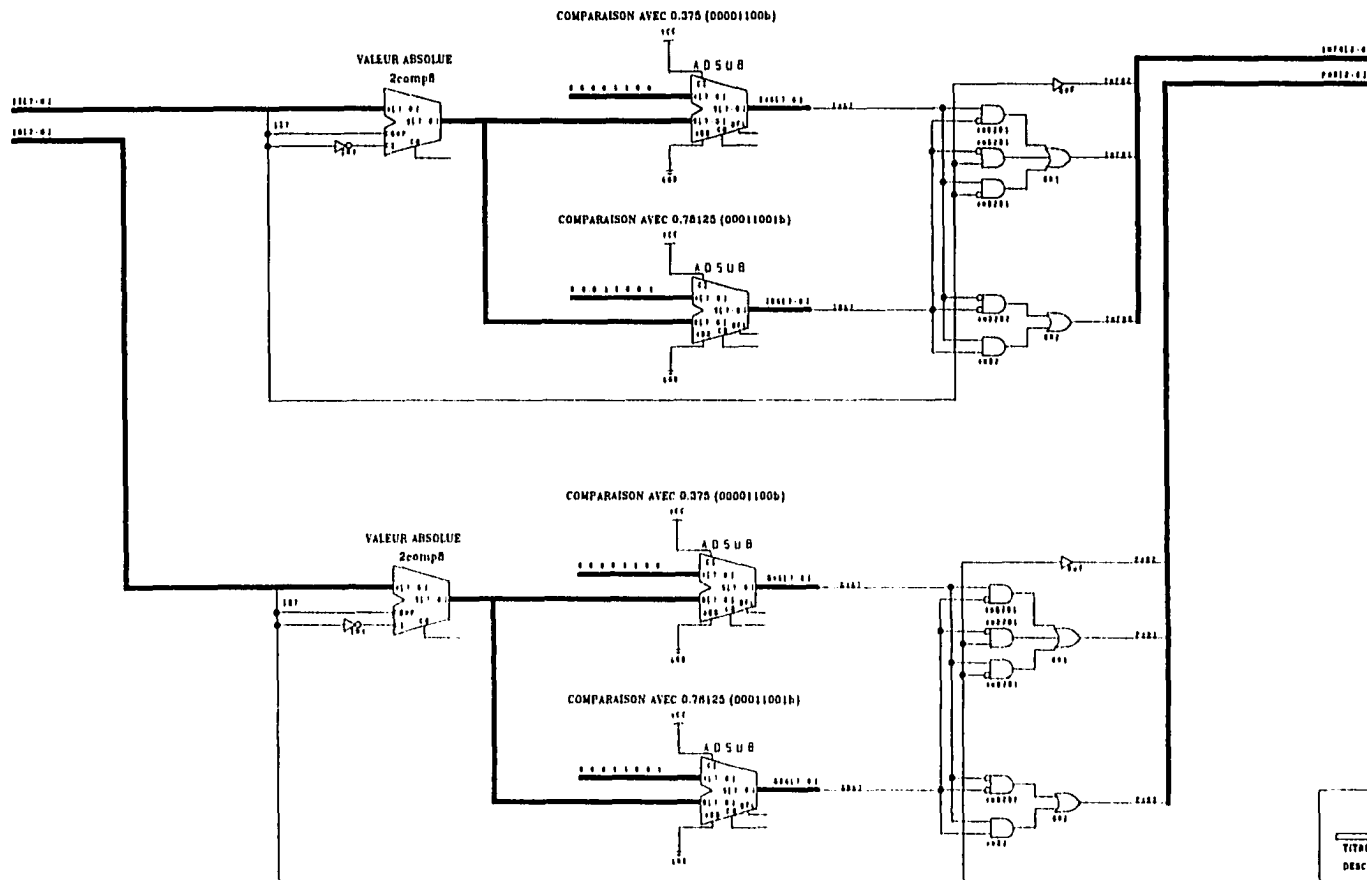
QUANTIFICATION DES SORTIES :  $-2^2 + 2^1 + 2^0$

POINT DE LA CONSTELLATION A +/- 1 (00100000b / 11100000b)

VALEUR DE INFO[2:0]



VALEUR DE II[7:0]



LACINE	
ECOLE DE TECHNOLOGIE SUPERIEURE	
TITRE DU PROJET: RECDR/CPK SYSTEM	
DESCRIPTION DE LA PAGE: QUANTIFICATEUR	
CODE DE PROJET: PAJ_0001	NOM: QUANTIFICATEUR
CODE D'OBJET: SY_00101R2	DESINE PAR: Marc Lantier
	DATE: 02/04/2000

# DECODEUR A SEUIL A QUANTIFICATION DOUCE

QUANTIFICATION SUR 3 BITS :  $-2^2 + 2^1 + 2^0$

EQUATIONS :

$$B0 = I0$$

$$B1 = P0 + I-7 + I-10 + I-16 + I-18 + I-30 + I-31 + I-35$$

$$B2 = P7 + I7 + I-3 + I-9 + I-11 + I-23 + I-24 + I-28$$

$$B3 = P10 + I3 + I10 + I-6 + I-8 + I-20 + I-21 + I-25$$

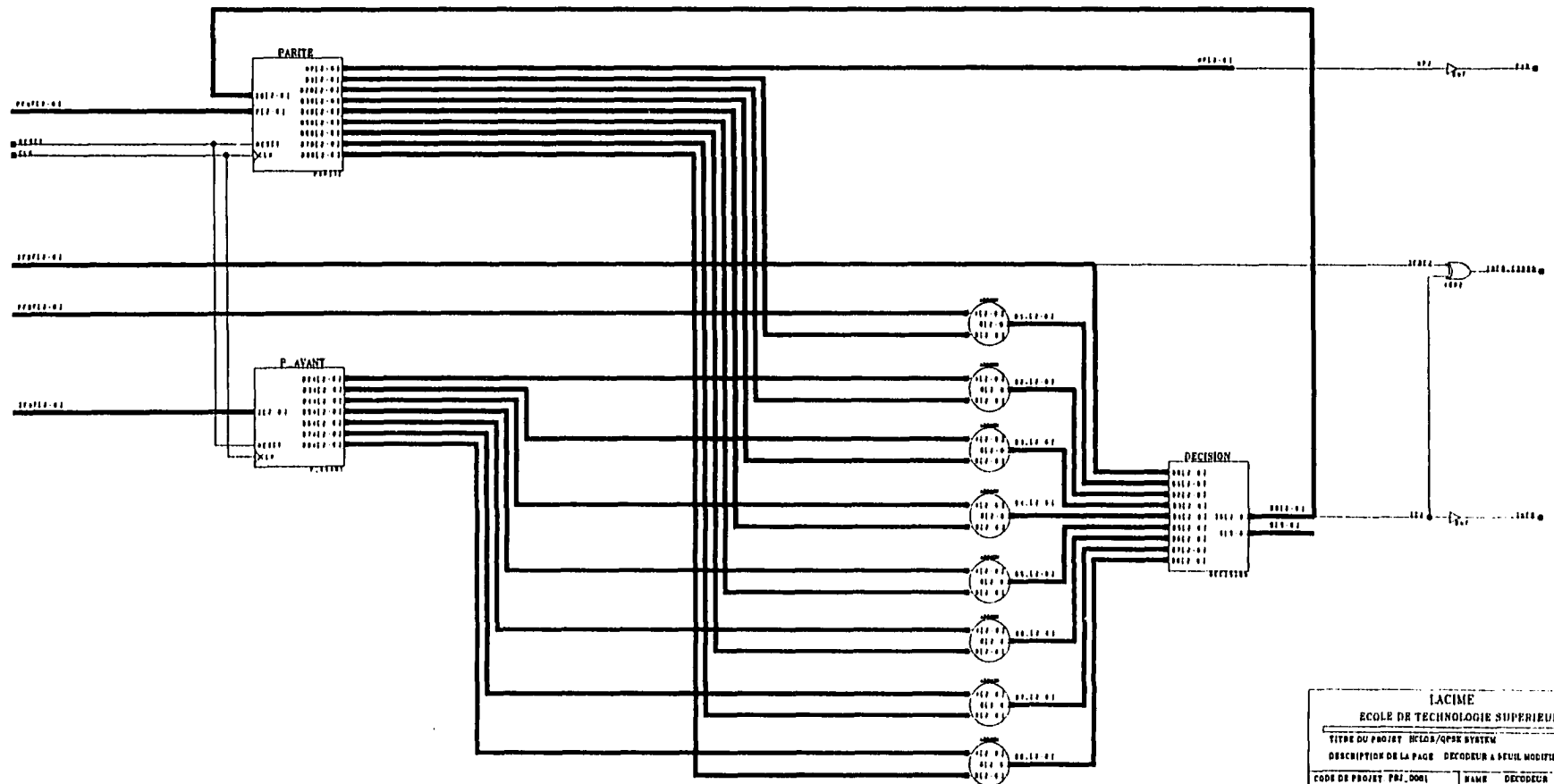
$$B4 = P18 + I8 + I9 + I16 + I-2 + I-14 + I-15 + I-19$$

$$B5 = P18 + I2 + I8 + I11 + I18 + I-12 + I-13 + I-17$$

$$B6 = P30 + I12 + I14 + I20 + I23 + I30 + I-1 + I-5$$

$$B7 = P31 + I1 + I13 + I15 + I21 + I24 + I31 + I-4$$

$$B8 = P35 + I4 + I5 + I17 + I19 + I25 + I28 + I35$$



LACIME	
ECOLE DE TECHNOLOGIE SUPERIEURE	
TITRE DU PROJET: HICLOS/QPSK SYSTEM	
DESCRIPTION DE LA PAGE: DECODEUR A SEUIL MODIFIE	
CODE DE PROJET: PA2_0001	NOM: DECODEUR
CODE D'OR/ET: RV_001618X	DESSINE PAR: MARY LAUSON
	DATE: 18/01/2000

# TRAITEMENT ET CALCUL DE LA PARITE

QUANTIFICATION SUR 3 BITS :  $-2^2 + 2^1 + 2^0$

EQUATIONS :

$$B1 = I-7 + I-10 + I-16 + I-18 + I-30 + I-31 + I-35$$

$$B2B = P7 + I-3 + I-8 + I-11 + I-23 + I-24 + I-28$$

$$B3B = P10 + I-6 + I-8 + I-20 + I-21 + I-25$$

$$B4B = P16 + I-2 + I-14 + I-15 + I-19$$

$$B5B = P18 + I-12 + I-13 + I-17$$

$$B6B = P30 + I-1 + I-5$$

$$B7B = P31 + I-4$$

$$B8B = P35$$

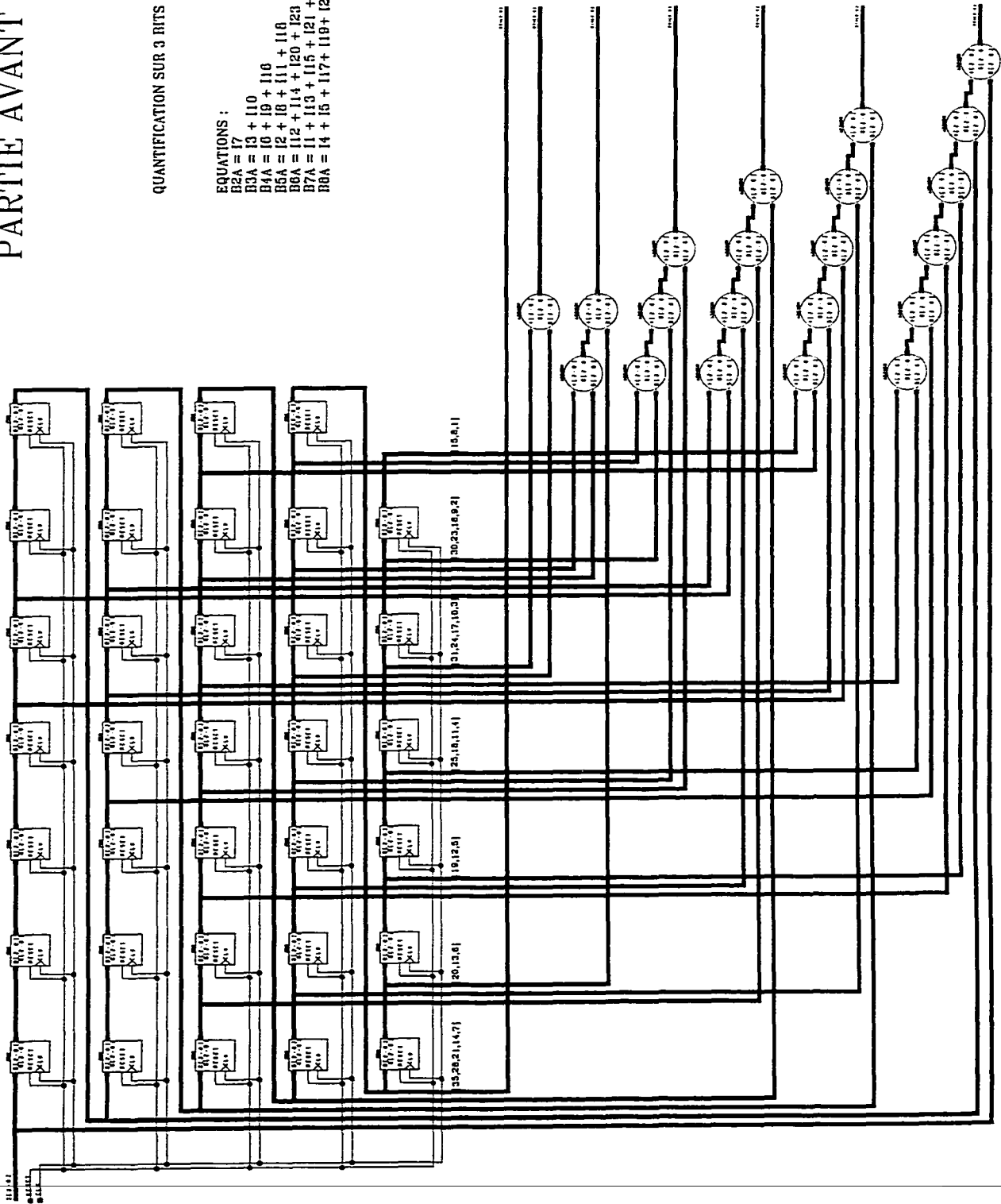
LACINE	
ECOLE DE TECHNOLOGIE SUPERIEURE	
TITRE DU PROJET : HCLDS/OPAK SYSTEM	
DESCRIPTION DE LA PAGE : TRAITEMENT DE LA PARITE	
CODÉ DE PROJET : P31,0061	NOM : PARITE
CODÉ D'OBJET : HY_001018	DESSINÉ PAR : MARY LAUREN
	DATE : 20/01/2000

# PARTIE AVANT DU DECODEUR

QUANTIFICATION SUR 3 BITS :  $-2^2 + 2^1 + 2^0$

EQUATIONS :

- B2A = 17
- B3A = 13 + 110
- B4A = 10 + 19 + 110
- B5A = 12 + 18 + 111 + 116
- B6A = 112 + 114 + 120 + 123 + 130
- B7A = 11 + 113 + 115 + 121 + 124 + 131
- B8A = 14 + 15 + 117 + 119 + 125 + 126 + 135



# ADDMIN A 2 ENTREES

QUANTIFICATION SUR 3 BITS :  $-2^2 + 2^1 + 2^0$

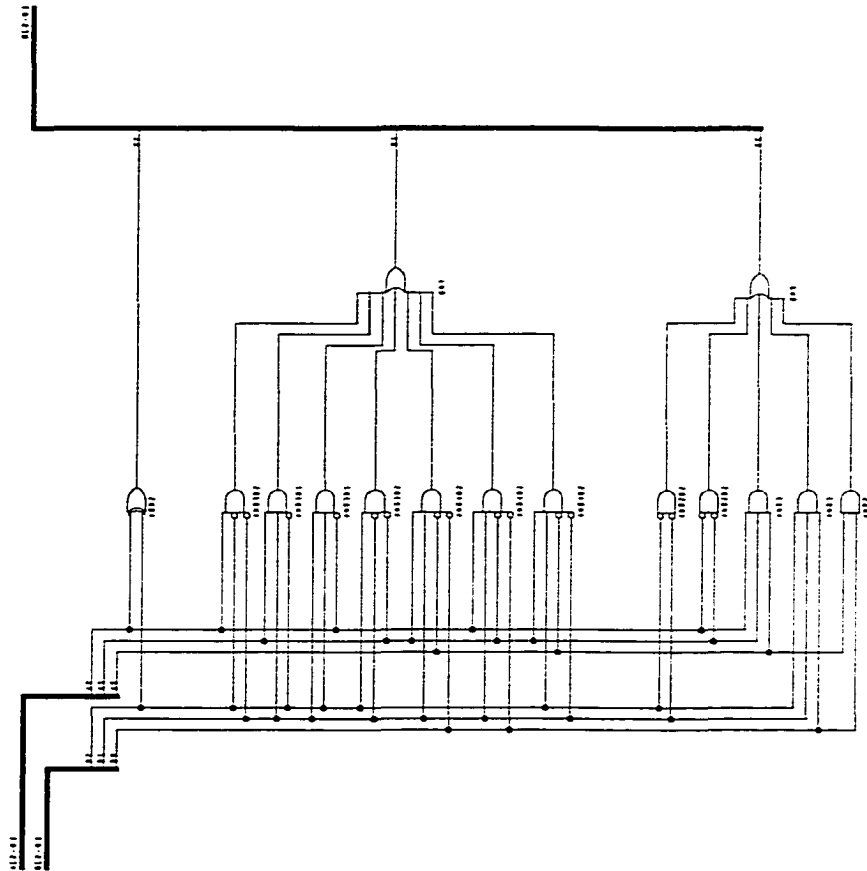
028180			TABLE DE VÉRITÉ									
A1A0	000	001	011	010	110	111	101	100	000	001	011	010
000	xxx	xxx	xxx	xxx	xxx	xxx	xxx	xxx	xxx	xxx	xxx	xxx
001	x1x	001	001	001	111	111	111	111	xxx	xxx	xxx	xxx
011	x1x	001	011	010	110	111	101	xxx	xxx	xxx	xxx	xxx
010	x1x	001	010	010	110	111	110	xxx	xxx	xxx	xxx	xxx
110	xxx	111	110	110	010	001	010	xxx	xxx	xxx	xxx	xxx
111	x1x	111	111	111	001	001	001	xxx	xxx	xxx	xxx	xxx
101	xxx	111	101	110	010	001	011	xxx	xxx	xxx	xxx	xxx
100	xxx	xxx	xxx	xxx	xxx	xxx	xxx	xxx	xxx	xxx	xxx	xxx

EQUATIONS :

$$02 = A2 \odot 82$$

$$01 = A2B2B1 + A1B2B1 + A1B2B1 + A1B2B1 + A1B2B1 + A1B2B1 + A1B2B1 + A1B2B1$$

$$00 = B2B1 + A2A1 + A2A1A0 + B2B1B0 + A0B0$$



ECOLE DE TECHNOLOGIE SUPERIEURE  
 TITRE DU PROJET : ADDMIN A 2 ENTREES  
 DEPARTEMENT DE LA PAGE : ADDMIN A 2 ENTREES DE 3 BITS  
 CODE DE PROJET : 0001  
 CODE D'OBJET : 0001  
 DATE : 1/04/2000  
 PAGE : 1/04/2000

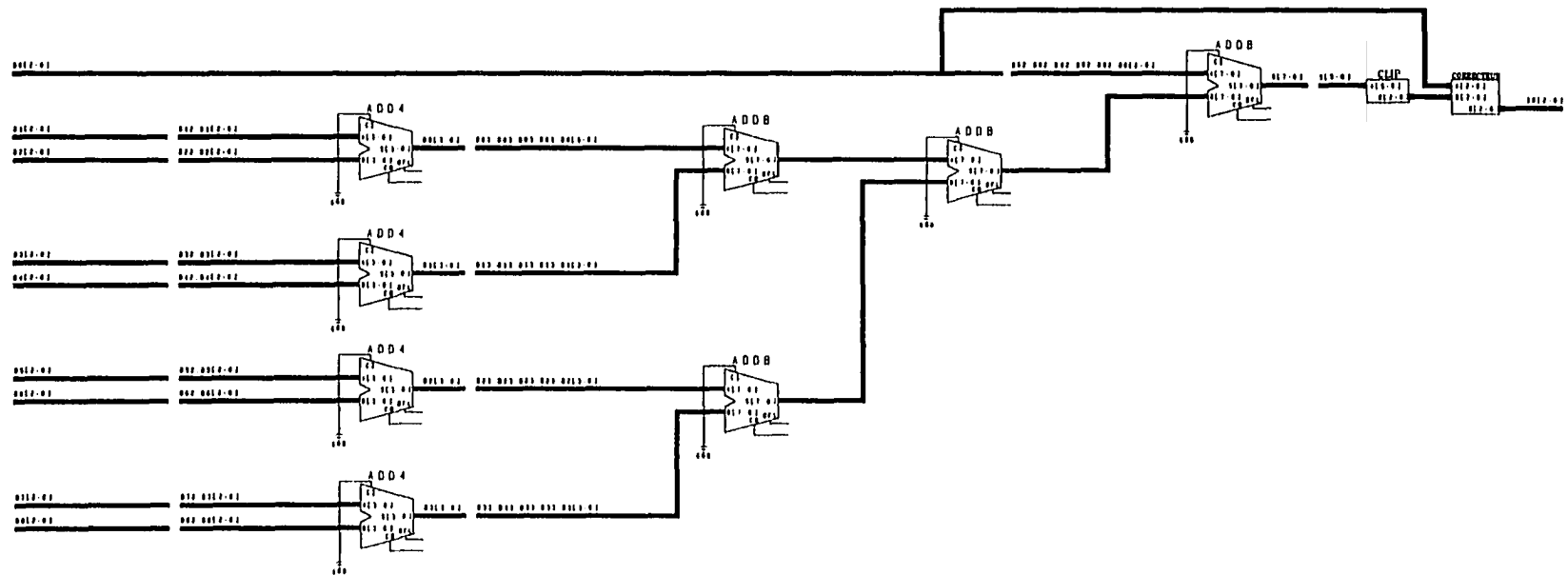
# BLOC DE DECISION

QUANTIFICATION SUR 3 BITS :  $-2^2 + 2^1 + 2^0$

LE RESULTAT DE LA SOMMATION EST LIMITE ENTRE -3 ET 3

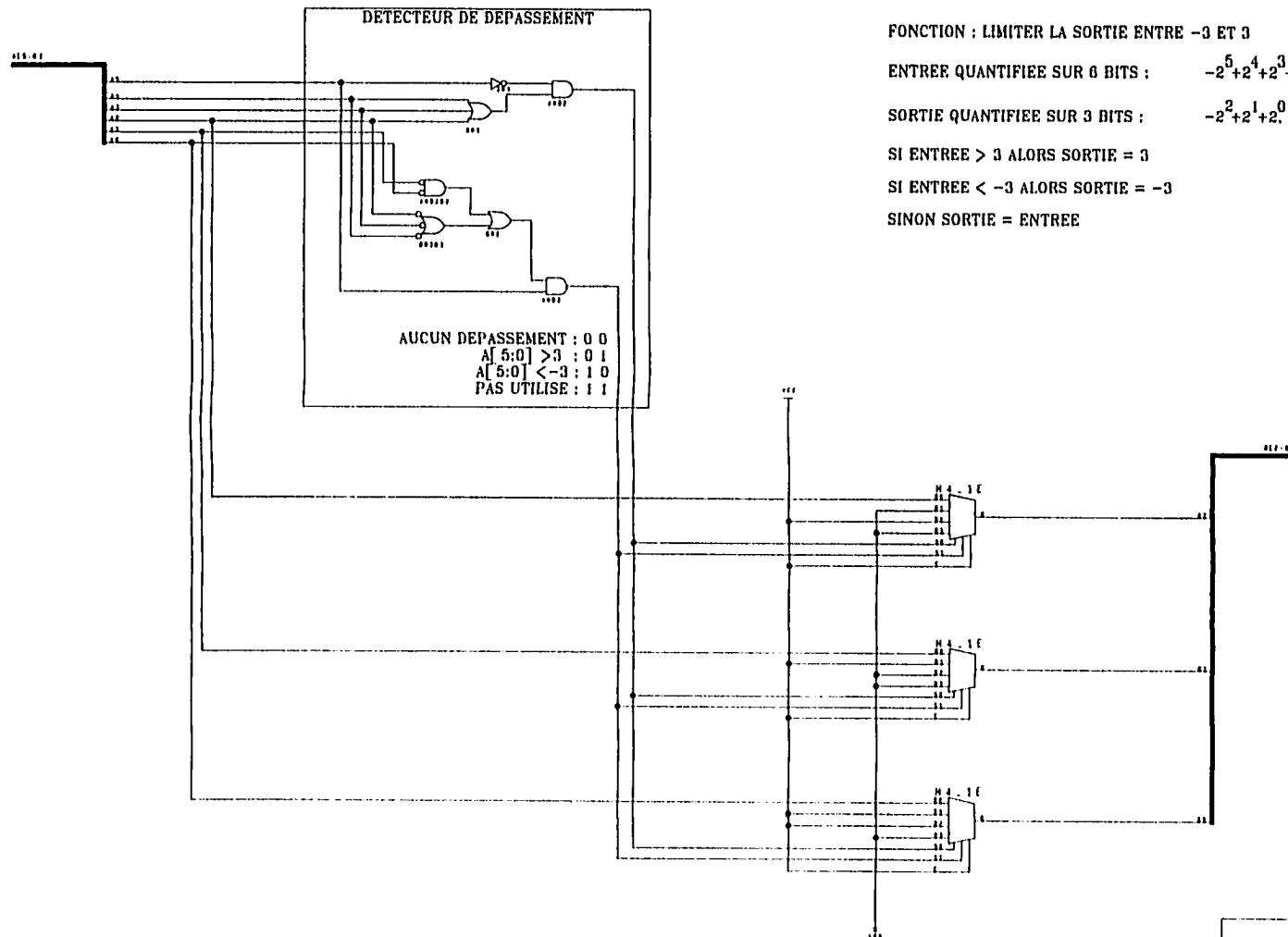
SI LE RESULTAT DE LA SOMMATION EST 0 ALORS SORTIE = R0[2:0]

SINON SORTIE = SOMME QUANTIFIEE



LACIME	
ECOLE DE TECHNOLOGIE SUPERIEURE	
TITRE DU PROJET	NCLOS/QPSK SYSTEM
DESCRIPTION DE LA PAGE	SOMME DE QUANTIFICATIONS
CODE DE PROJET	FRI_0001
CODE D'OBJET	HY_00101010
NOM	DECISION
DESIGNE PAR	Mauro Lamine
DATE	25/01/2000

# CLIP



LACINE	
ECOLE DE TECHNOLOGIE SUPERIEURE	
TITRE DU PROJET : HCLAB/GPSK SYSTEM	
DESCRIPTION DE LA PAGE : BLOC LIMITANT LA SORTIE	
CODE DE PROJET : PRI_0001	NOM : CLIP
CODE D'OBJET : VI_COUBINE	DESSINE PAR : MAYA LAURIN
	DATE : 25/01/2000

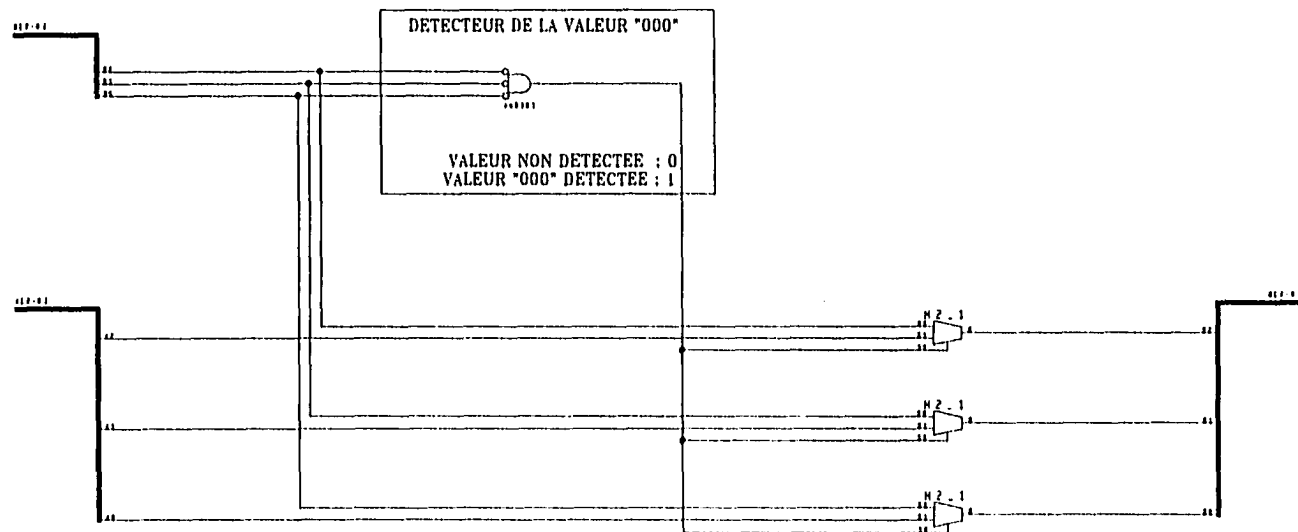
# CORRECTEUR

FONCTION : EXCLURE LA VALEUR 0

QUANTIFICATION SUR 3 BITS :  $-2^2 + 2^1 + 2^0$

SI ENTREE B = 0 ALORS SORTIE = ENTREE A

SINON SORTIE = ENTREE B



LACINE	
ECOLE DE TECHNOLOGIE SUPERIEURE	
TITRE DU PROJET : MICRO/PROCESS SYSTEM	
DESCRIPTION DE LA PAGE : EXCLURE LA VALEUR 0	
CODE DU PROJET : PRJ_0001	NOM : CORRECTEUR
COD D OBJET : RV_001010	DESSINE PAR : Mary LAUREN
	DATE : 25/01/2000